

Contents

Welcome to DLG Professional	1
What is DLG?	1
What is A BBS?	1
What You Can Do With DLG	2
What Is Expected Of A DLG SysOp	2
How to use this manual	2
Installation of DLG Professional	5
Check your DLG Professional package	5
Check Your Equipment Setup	5
Attention Workbench 1.3 Users	5
Attention ARP Users	5
Register your Product	5
Before you Install DLG	6
Installation from WorkBench:	6
Installation from CLI:	6
How to use the DLG Installation Software	7
Upgrading DLG from a previous version	8
Trouble-shooting the installation	8
If You Forget Your Password	9
If You Mess Up Your Account	9
Terms & Concepts	11
Overview of DLG	12
Text based	12
User Accounts	12
Group Accounts	12
SIGs	12
Overall Structure	12
Language Configurable	13
DLG Menus	13
Command Stacking	13
Hot Keys	14
Smart Input for SysOps	14
TPT_Handler	14
TPTCron	14
TPTRM	15
User-Levels	15
User Account Basics	15
Message and File Area Basics	16
DLG Startup and Quick Tour	17
General preparations:	17
Amiga CLI	17
Starting DLG	17
Logging in to DLG	17
A Quick Tour of the DLG System	18

Required Steps to Configure Your System

23

Logging in to DLG	23
SysOp Menu	23
Port Configuration	24
Global Configuration	24
Modem Configuration	26
Display Configuration	28
DLG Ports	30
Summary	32
Example Section	32
Reference	34
Port Configuration Overview	34
Modem Configurations	34
Global Configurations	36
Display File	38
Port Configuration	41
The Port Configuration Editor	43

DLG Message Areas

45

DLG Message Areas — An Overview	45
Tutorial — How to Create a Message Area	45
Entering the DLG Message Base	48
Message Area 1	48
Tutorial - Entering a Message	51
Tutorial - Editing An Existing Message Area	53
User Access to Message areas	53
Conclusion and Tips	54
Example Section	55
Reference Section	56
DLG Message Areas	56
Other Message Area Information	60
Message Translation	60
Area Message Filters	61
Welcome Text	61
User Access to Message Areas	61
DLG Message Software	63
Message Area Commands	64
Miscellaneous Notes	69

File Areas

71

Tutorial — Creating a File Area	71
The Philosophy Behind DLG File Areas	72
Tutorial — Create A File Area	72
Entering the DLG File Base	74
Tutorial - Uploading a File	81
Tutorial - Commenting a File's Description	82
Tutorial - Editing a File	83
Tutorial - Downloading a File	85
File Area Maintenance Overview	86
Tutorial - Batch Uploading Files	88
Tutorial - Uploading files from a CD ROM	90
CD ROM guidelines	91
CD ROM Side Effect	93
Tutorial - Modifying or Deleting a File Area	93

Conclusion and Tips	94
Example Section	95
Reference Section	96
DLG File Areas	96
Other Options That Affect DLG File Areas	98
DLG File Area Text and Batch Files	98
File Area Maintenance Editor	99
User Access to File Areas	100
DLG File Software	101
Miscellaneous Notes	103
DLG Text, Batch and Configuration Files	105
%Switches	105
DLG Text Files	107
Optional Text Files	109
DLG Language files	111
Character Translation Sets	112
Script Files	113
Essential Script Files	113
Optional Script Files	114
FidoNet Script Files	116
DLG Configuration Files	118
DLG Menus	121
DLG Custom Menus	121
Format of a Custom Menu Text File	122
Tutorial: Creating a Custom Menu	123
Other features of DLG Menu Sets	125
DLG Configurable Menus	125
Reference Section:	126
DLG SysOp Editors	137
The Main SysOp Menu	137
User Account Edit	138
Group Account Edit	141
Port Configure	141
Network Mail Configure	142
Configure Menus	142
File Area Maintenance	142
General Configuration	143
Log Configuration	143
Editor Configuration	143
Protocol Configuration	144
Archiver configuration	146
Character Set Configuration	147
File Path Configuration	147
SIG creation/edit	148
DLG Command File Reference	149
Network Mail Overview:	189
FidoNet:	189
FidoNet Software	190
UseNet Network:	190

DLGMail	193
What is DLGMail?	193
Structure	193
Major Non-DLGMail programs used in connection with DLGMail	194
Setting Up FidoNet	194
Creating Areas In DLG	194
DLGMail Installation	195
Paths and assignments used with DLGMail	195
The FIDO:DLGMail.CFG file	197
Settings	197
Archiver Configurations	205
Using the DLGMail Executable	206
The Built-In ARexx Commands	206
Environment Variables	208
DLGMail.ARE file	209
DLGNet/DLGBundle Configuration and Control Files	211
Messages, Packets, Bundles and Flow Files	212
Wildcards with DLGNet and DLGBundle	214
The DLGMail.BUN Control File	214
The DLGMail.MRT Routing File	216
The DLGMail.PNT Routing File	218
The Process Cycle	218
The DMC CHANGE Commands	219
DMC and DLGMail Overview	221
Additional Support Executables	226
Servicing Points	232
Configuration	234
Editing Scripts and Configs	234
Requesting a Permanent Fido Address	237
Requesting an Echomail Hub	237
Finding Your Comfort Zone	238
Nodelist Compiling	239
TrapDoor	245
Introduction	245
FidoNet	245
Mailer Operation	245
Nodelist	245
Accounting	246
Security	247
Trapdoor Setup	249
Example Modem Settings	250
Configuration Commands	251
Keywords	252
TrapDoor with ARexx	269
TrapDoor without ARexx	270
Appendix	279
%Switches	279
Built-In Functions	281
Local Variables	287
Index	289

Welcome to DLG Professional

DLG Professional is the "Bulletin Board Operating System" for the Commodore Amiga. DLG provides the software tools and environment necessary to create a flexible electronic bulletin board system.

What is DLG?

The disk operating system (AmigaDOS) of your Amiga consists of a number of commands which are loaded into memory and executed when needed. Each command performs a specific task - copying a file, playing a sound, displaying a picture, and so on. DLG is very similar to AmigaDOS in concept. We took all of the various tasks performed by a bulletin board system and broke them down into small programs, each of which performs a specific task. DLG's commands include those that write messages, send files, perform maintenance, and so on. Like AmigaDOS, DLG supports batch files (lists of commands to execute in sequence), command line arguments, etc.

It is this modular approach that makes DLG unique. Other bulletin board packages usually consist of a single, all-purpose program that resides in your Amiga's memory. Even though the code that sends and receives files is dormant while a message is being written, it is still there, taking up precious RAM. DLG avoids this waste of resources by loading only that code which must be present for a given task. DLG frees up your RAM to allow you to multi-task other programs with your bulletin board system.

DLG excels in another area - multi-tasking within itself. Because of the modular nature of DLG, and because it is "built" on a standard AmigaDOS shell, all of the DLG commands can multi-task with each other, allowing easy creation of multi-line systems. One user can be receiving files, while another plays an on-line game, while still another reads and replies to mail!

We designed DLG to be the most powerful and complete software of its kind. We also made DLG easy to install, operate, and maintain, making it an excellent choice for professional or commercial settings. It has an easy-to-use interface and efficiently uses your computer's resources, making it ideal for club or hobby use.

What is A BBS?

Electronic bulletin board systems are generally known by the term "BBS". A BBS can provide a number of services:

- public and private communication between the people who use the BBS.
- public and private exchange of computer files between the users of the BBS.
- forums for discussion of current news, entertainment, and debate topics.
- networked communications with users in other geographic locations.

Bulletin board systems available for most computers provide most or all of these functions. DLG provides these functions, and offers you more. Features which enhance it's abilities for the professional user include:

- on-line conferences of multiple callers.
- the ability to interface with more than one network mail service, such as UseNet or FidoNet.
- group or department accounts with coordinators for group accounts.
- public bulletins and announcements.
- user accounts with individual message and file areas for each user.
- private and secure access to services.

These features make DLG a professional choice for heavy-duty applications. DLG also has a number of features which makes it an excellent choice for home, hobby, or club use. DLG offers low memory usage, and low overhead, allowing the software to run on modest setups. No longer do you have choose between running a bulletin board or enjoying your computer - with DLG you can have both!

What You Can Do With DLG

Using the DLG software, you will be able to create and maintain a bulletin board system of any size and complexity. You can run a small "hobby" BBS that requires a minimum of maintenance and operator intervention, or you can run a mammoth information-service type BBS with multiple phone lines, and gigabytes of files and messages available to your users. The choice is yours. The DLG software will not limit your horizons.

What Is Expected Of A DLG SysOp

Like any piece of complex software, the DLG system will take time to learn. If you follow the guidelines in this manual, you will gain a good grounding in what it takes to make DLG work for you, and what you need to do to make DLG function smoothly.

You will be learning some new concepts, and you will have to spend some time at first, working with and tweaking your DLG system. We have provided the tools you need to create the bulletin board system of your dreams, but there are some ingredients we could not include in the box - your creativity, interest and time. The more time and effort you invest in DLG, the more it will reward you.

You should be familiar with the Amiga computer, the CLI environment, and the creation, editing, and execution of AmigaDOS script files. You should also be familiar with telecommunications terminology, and the various techniques and tools commonly used in Amiga telecommunications.

If you require additional material on AmigaDOS commands, and the AmigaDOS scripting language, please refer to the Commodore-Amiga Inc. publication "*The AmigaDOS Manual 3rd Edition*," published by Bantam Books, 1987.

How to use this manual

This manual has been designed to help you get your system installed and configured quickly. However, before you begin to work with the DLG software we recommend strongly that you read the chapter called "Overview of DLG Concepts." Many of the ideas introduced in that chapter will be needed as you work through the tutorials.

A large part of the initial part of the manual is devoted to "hands-on" tutorials. These tutorials will guide you through the early stages of setting up your DLG system. It is important that you use this manual and work through each of the tutorials while you are sitting at your Amiga. You will not profit as much from merely reading through the manual.

Here is a general overview of the structure of the manual:

- Introduction
- Installation of software
- Explanation of terms & Overview of DLG concepts
- Startup and Quick Tour
- Required steps to setup DLG
- DLG Message Areas
- DLG File Areas
- Personalizing your DLG System - Text, Batch, and Configuration files

- Menu Configuration
- Sysop Editors
- Reference Section
- Network Mail Overview
- DLGMail
- TrapList
- TrapDoor

Since much of the DLG software contains descriptive prompts for user input, the manual will strive, wherever possible, to point you in the proper direction. The software itself will guide you, step by step, through the operations required.

Installation of DLG Professional

Check your DLG Professional package

Your DLG Professional package should contain the following:

- Three numbered disks
- The DLG Professional Manual
- A registration card

Check Your Equipment Setup

You can install and run DLG Professional on any model Commodore Amiga computer system. Your computer should have the following items:

AmigaDOS version 1.3 or higher.

At least 512 Kilobytes of RAM. If you plan on using FidoNet or UseNet services, you should have a minimum of 1 Megabyte of RAM — more is recommended.

A hard drive with at least 5 megabytes of free space. You will need a minimum of 3 megabytes to install the software, and additional storage space for message, file areas, and user accounts.

A modem connected either to the Amiga's serial port or installed internally.

A telephone line connected to the modem.

If you plan on creating a large system with DLG, you should consider the following extra equipment and software:

Additional memory, especially if you are planning on implementing FidoNet or UseNet services with a multiple line system, or if you wish to multi-task other software with your DLG system.

Additional hard drive space for message and file areas.

Additional serial ports, modems, and telephone lines.

Attention Workbench 1.3 Users

If you have not already done so, TelePro Technologies encourages every Amiga owner to upgrade their Amiga to use Workbench 2. DLG version 1.0 *will* run under AmigaDOS version 1.3 or higher. Future versions of DLG will require the use of AmigaDOS version 2.04 or higher. We feel that Workbench 2 significantly adds to the enjoyment of using your Amiga computer. More importantly, Workbench 2 provides a richer programming environment, and more operating system features. Future versions of DLG will take advantage of the new operating system features. This will make it incompatible with older versions of Workbench.

Attention ARP Users

DLG will not function with the AmigaDOS Replacement Project (ARP) "NewCLI" command installed. If you are running ARP, you will need to replace the ARP CLI command with the standard Commodore-supplied NewCLI command. If you use the ARP NewCLI command, your Amiga system will fail or "Guru" when you attempt to log-in to your DLG system.

Register your Product

To receive registered customer services, complete and return the registration card included in the product package.

If you are missing any item from your DLG package, please contact our customer service department by voice or fax at 403-341-7826, or call our support BBS at 403-347-3262 (up to 14,400 bps, HST Modems) or 403-347-3269 (up to 14,400 bps, V.32bis Modems).

Before you Install DLG

1. Make sure that you have sufficient hard drive space to hold all the DLG directories and files. You should have at least three megabytes of hard drive space for proper installation of the software. Note that this is a minimum requirement for installing the software. To run a BBS of any significant size you will require additional space for message and file areas.
2. The DLG software is organized into several directories. If you would like all of these directories grouped inside of a single directory on your hard drive, you will need to create that directory PRIOR to running the installation utility. For example, if you want the installation program to install all of DLG into a directory called "TelePro" on your Work: partition, you would need to execute the following in your CLI:

```
makedir Work:TelePro
```

Then you would be able to instruct the installation utility that you wish the software to be installed there.

3. Make sure that each of the installation disks is write-protected before you insert them into your disk drive. This will not only protect them from accidental erasure, but will also guard against possible virus infection of the installation disks.
4. Make backup copies of your DLG Professional disks. Your license agreement authorizes you to make one backup copy of each DLG installation disk. For information about copying disks, refer to your Amiga documentation.

Installation from WorkBench:

1. Insert *Disk1* into any floppy drive on your Amiga. If you have more than one floppy drive, insert *Disk2* into the second drive. If you have 3 floppy drives, insert *Disk3* into the third drive. If you only have one floppy drive, you will get system requesters asking you for the disks as they are needed.
2. Double click the *Disk1* icon to open up the disk window.
3. Double click the *DLG_Install* icon.
4. Go to the section "How to Use the DLG Installation Software" below.

Installation from CLI:

1. Insert *Disk1* into any floppy drive on your Amiga. If you have more than one floppy drive, insert *Disk2* into the second drive. If you have 3 floppy drives, insert *Disk3* into the third drive. If you only have one floppy drive, you will get system requesters asking you for the disks as they are needed.
2. Type: "cd Disk1:" at your CLI prompt and press RETURN
3. Type: "execute CLI-Install" at your CLI Prompt and press RETURN
4. Follow the installation instructions below.

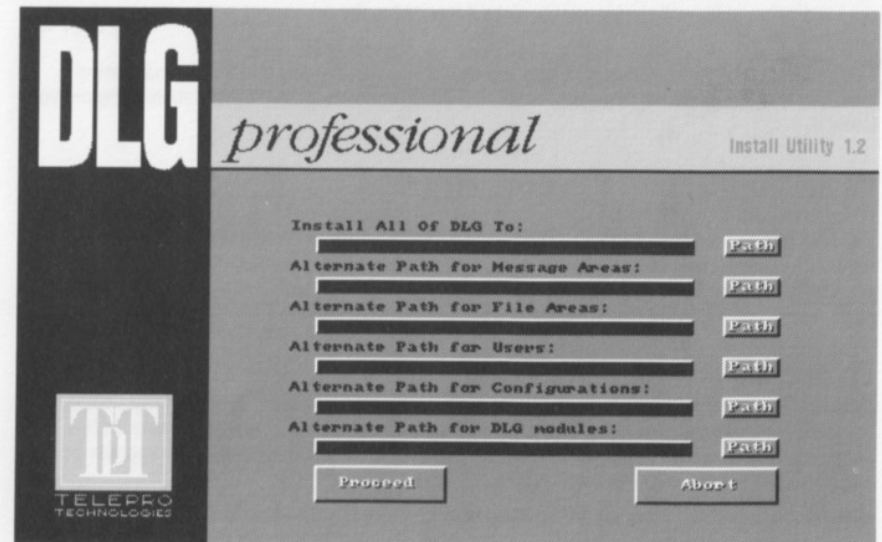
How to use the DLG Installation Software

DLG is a complex system that requires proper installation. Many directories must be created, and many files must be located properly within those directories for DLG to function. The DLG Install program provides you with an easy method of performing the installation. From time to time you will have the opportunity to upgrade an existing DLG installation with a newer version of the software. You will use this very same installation utility to install software upgrades.

If you are installing DLG for the first time, you need only decide where you would like the software installed. Since DLG is modular in nature, it does not require installation on a single volume. The DLG programs can be in one place, the file areas in another, the message areas in another, and so on. The installation utility provides you with an easy method for placing the various parts of the DLG package where you would like them installed.

There are some things to think about before performing the installation. If you only have one hard drive partition, then you may as well place all of the DLG software in a single directory. If you have several partitions, or more than one physical hard drive, then you have the choice of placing the various parts of the software where you will get the maximum benefit of your expanded storage space. The DLG programs and configuration files do not change size to any great degree when you are running the software, so you could place them on a volume with at least 2 megabytes of room. Your file areas will probably take up the largest amount of space, depending on the kind of system you are planning on, so you should consider placing them on the volume with the most free space. Your message areas and user files will take a medium amount of space — figure on about 100K per message area, and 1K per user.

The installation utility presents you with a screen containing a number of text requesters. You are to type paths into these text requesters, indicating where the various parts of DLG are to be installed. You can either type a pathname directly into the text requesters that you see, or you can interactively select the path from a special requester listing your volumes, assigns, and directories, by pressing the PATH button located to the right of each text requester.



If you only have one hard drive or partition, simply place the pathname of the directory where you would like DLG to be installed in the topmost text requester, or interactively choose the path using the PATH button.

If you have several hard drives or partitions, you can enter alternative paths for each of the parts of DLG in the other text requesters. You do not have to fill in every requester. If you leave a requester blank, the installation software will use the main path you entered in the topmost requester to install that particular part of DLG. For example, if you wanted to place all of the DLG software, configuration files, user accounts, and message areas on your WORK: partition, but you wanted to place your file areas on a different volume, you would simply put WORK: in the topmost text requester, and place the name of the other volume in the file area text requester, and leave all of the other text requesters empty.

The installation utility will prompt you to provide your name and a password. As part of the installation procedure, the software will create a SysOp account for you. Be sure to use a password that you can remember.

The installation utility will ask you for the number of ports that you will be running. The minimum number of ports is two — one for your local log-in sessions, and one for the serial port connection. If you have additional serial ports, modems, and telephone lines, then add the number of additional ports you have to the minimum two ports provided for. For example, if you are adding the Commodore 2232 multi-serial card to your Amiga to provide an additional 7 serial ports for use with DLG, then your total number of ports will be 9.

The installation utility will ask you if you want to have the FidoNet utilities installed. We suggest that you answer "YES" even if you do not want to interface with any FidoNet-type network right now. It is not impossible to add these utilities to your DLG installation after it has been set up, but the task will be easier if you go ahead and install them now. The FidoNet utilities will remain dormant until they have been properly configured. If you are positive that you will never interface your DLG system with a FidoNet-type network, then answer "NO" to this question. It will save you a bit of hard drive space.

Once you have provided this information, the installation utility will proceed. If you have one floppy drive, a requester will prompt you at some points to swap disks. If you have multiple floppy drives, the installation utility will simply use the disks if inserted in the drives, and not prompt you to swap them.

The installation utility has on-screen help and instructions to help you through the installation procedure. It is actually very painless — you will have your DLG system up and running before you know it.

Upgrading DLG from a previous version

If you are upgrading an existing DLG installation, the installation software will automatically detect this fact, and proceed accordingly. It does this by testing for the existence of certain "assigned" DLG volumes. Therefore, it is important to the upgrade procedure that you have started DLG before you attempt to perform the upgrade. If you attempt an update when DLG is not active, the update will fail!

Trouble-shooting the installation

Occasionally we hear reports from customers who have had trouble with their DLG installation. This is usually because they have aborted the installation part-way through and started over. In this situation, the installation utility can be "fooled" into thinking that it is to perform an upgrade on the software, and not perform a full installation.

If, for any reason, you abort the installation procedure once it has started to create directories and copy files, you will need to re-boot your computer to remove assignments that the installation software made. You will also need to remove any directories or files that it has installed before you attempt the installation a second time.

If you have ARP installed on your system, be aware that the ARP commands have been known to cause the installation to fail.

If You Forget Your Password

If you forget the password that you gave the installation software, all is not lost. You can find out the password by using a couple of commands from your CLI:

```
cd user:your_name
```

(substitute your name here, but put underline characters "_" instead of spaces. If your name is Gene Hamm then you would type CD User:Gene_Hamm)

```
type user.data opt h
```

You will see a listing on the right-hand part of your screen of some text mixed in with binary characters. The very first text that you see is your password.

If You Mess Up Your Account

If you accidentally do something to your SysOp account so that you cannot log-in, or can no longer access the SysOp menu, here is a quick fix:

Insert *Disk1* of the DLG installation disks in DF0:

Type the following into a Shell:

```
Disk1:Install_Files/adduser "Your Name" "password"
```

and press RETURN.

Substitute your own name, and the password you want. Be sure to put the name and password in quotation marks. The name you use here cannot be the same as the account that you have messed up. Spell it differently, or put your initial in, or do something to differentiate it from the other account.

You will now be able to log in under the new name and password.

If you have any questions or need assistance, please contact your local distributor or call 1-800-4-A-SECURITY.

If You Forget Your Password

If you forget the password for the system, you can reset it. To do this, you must first enter the correct password for the system. Once you are at the main menu, press the F2 key. This will take you to the password reset screen.

On the password reset screen, you will be prompted to enter a new password. Please make sure that the password is at least 8 characters long and contains at least one uppercase letter, one lowercase letter, and one number.

After you have entered a new password, you will be prompted to confirm it. Once you have confirmed the password, the system will save the new password and you will be able to log in with the new password.

If You Lose Your Access Card

If you lose your access card, you should report it to your supervisor immediately. Your supervisor will then contact the security department to deactivate the card.

Once the card has been deactivated, you should stop using it immediately. If you continue to use the card, you may be subject to disciplinary action.

You should also be aware that if you lose your access card, you may be required to pay a replacement fee.

If you have any questions or need assistance, please contact your local distributor or call 1-800-4-A-SECURITY.

The System Features

The system features a variety of advanced security features, including motion detection, video surveillance, and access control. The system is designed to provide comprehensive protection for your facility.

For more information, please contact your local distributor or call 1-800-4-A-SECURITY.

The system is also compatible with a variety of other security systems, including fire alarms and intrusion detection systems.

For more information, please contact your local distributor or call 1-800-4-A-SECURITY. The system is designed to provide comprehensive protection for your facility.

Terms & Concepts

This chapter contains an explanation of some of the terms used in this manual. This chapter also presents a brief overview of the structure and concepts important to a full understanding of the DLG software. Please take the time to read this chapter because many of the concepts and ideas presented here will be referred to in the later chapters that contain tutorials. The information presented here is mostly non-technical, and is intended to give you a solid foundation for the information in later chapters.

SysOp - This stands for **System Operator**, and refers to the person running a bulletin board system or other information service. This is you! A SysOp has access to all features, commands, areas, and abilities on a DLG system.

Co-SysOp - This is a user on the system who is given special access by the SysOp. A Co-SysOp usually performs the same duties as a full SysOp, but only in certain areas of the system. For example, you may create areas on your system for Macintosh users, and assign a Co-SysOp to help manage the Macintosh message and file areas for you. A Co-SysOp should not be given access to the SysOp menu, or user account editor, unless you trust and know them very well.

User - This is a person who calls into a bulletin board system. A user has only the amount of access that the SysOp has allowed. Some features, commands, areas, and abilities will be unavailable to a user, unless the SysOp specifically allows them.

Port - A port is an entry way into the DLG system. On a base DLG installation there are two ports - one for "local" or keyboard entry, and one for "remote" or modem entry. On a multi-line system, you would have additional "remote" ports - one for each additional modem/telephone line. It is important to understand the concept that there are multiple entry ways in the DLG system - one for each port that is active. Some of the things you can do with your DLG system will affect all ports equally, and some will only affect a single port. It is quite possible to create two radically different BBS systems with a single DLG installation - which one users see depends on the port they enter the system through.

Menu - The main form of interface between DLG and users is the menu. DLG prints lists of letter commands in a text menu format. To select a command from the list, you type the letter corresponding to the command that you want, and press the RETURN key on your keyboard. DLG will respond accordingly. DLG menus are "smart" menus, and only show options which are available, and make sense, at any given moment.

FidoNet - A protocol created and developed by Tom Jennings of Fido Software that facilitates the exchange of messages between different bulletin board systems. FidoNet allows for the exchange of person-to-person messages (NetMail) and for public message areas that are shared across many computer systems (EchoMail). FidoNet is used in this manual in a generic fashion. While the original network is called "FidoNet", there are dozens of alternative networks which go under different names but still use the same software and protocols that FidoNet does. We use the term FidoNet to include all such networks, not just the original FidoNet network. DLG can interface with FidoNet through Maximillian Hantsh and Martin Laubach's TrapDoor and Steve Lewis and Ross Martin's DLGMail (both included with your DLG package) or with many other third party FidoNet utilities.

UseNet - A network for the exchange of messages and files that is primarily based on large government, university or corporate owned computer systems. DLG can interface with UseNet through Matt Dillon's UUCP software.

Overview of DLG

Text based

Unlike most Amiga programs, DLG is text based. It does not have the pull-down menus and gadgets which other Amiga software provides for user input. DLG communicates primarily through a text window. The reason for a text interface is that DLG is a telecommunications program, and text displays are the main medium for information in telecommunications. It presents the same interface whether you are logging into the DLG system on your local keyboard, or if you are calling your system from a terminal in another city.

User Accounts

Each user account in a DLG system has a directory contained in a logical volume named USER:. Contained in that user directory are various files which define the kind of account that user has, private message and files which may have been sent to that user, and various other pointers and data files.

Group Accounts

A group account is an account for a collection of users that you can define. A message can be sent to the name of a group, and all members of that group will receive a copy of that message. A group account consists of a Group-Op and regular members. A file sent to a group will be received only by the Group-Op.

SIGs

SIG stands for Special Interest Group. In DLG terminology, a SIG is a collection of related message or file areas that you can define. You define a SIG by creating it with a descriptive name, and indicating which message or file areas belong to that SIG. When a user selects a SIG, only those message and file areas that belong to that SIG will be visible.

Overall Structure

At its most basic level, a DLG BBS system consists of a main menu, a message base, a file base, a conferencing area, a SysOp maintenance area, and a collection of miscellaneous utilities. By using text menus and prompts, users can navigate around the DLG system, moving from one area to another to do and see various things.

The Main Menu is the crossroad of the DLG system. All other parts of DLG branch out from the Main Menu.

The message base contains a number of message areas, usually divided up by a topic or theme. Users are able to leave messages to each other in the various message areas. DLG also has a private message function. Unlike other BBS systems, private messages in DLG are held in the recipient's private mail area, not scattered all over the various public message areas.

Each public message area is contained in its own directory in a logical volume named MESS:, with each message contained in a separate file. Private messages are stored in each user's individual account directory.

The file base contains a number of file areas, usually categorized by type or kind of file. Users are able to upload files. You can define whether user uploads will go directly into each area, or to a central "validation" area, where you can check each file before allowing it to go on-line. You can define this on an area-by-area basis. This redirection of uploads to the validation area is totally

transparent to the user. A unique feature of DLG is that users are also able to upload files to each other privately. Users have individual private file areas in which they are able to receive files, up to a SysOp-definable size limit.

Each public file area is contained in its own directory in a logical volume named FILE:, with each file description contained in a separate file. The actual downloadable files in a file area can be optionally located in the file area directories, or on an alternate volume, or in any one of a series of globally defined volumes. Private files are stored in each user's individual account directory.

The DLG conferencing area allows users on a multi-line system to create various rooms and chat with each other. It is also possible for people in a conference to communicate privately with each other, without other people in the conference being aware of it.

The SysOp maintenance area contains many "editors" which allow the SysOp to create and edit message and file areas, modify user accounts, create group accounts, maintain message and file areas, and configure the DLG system in various ways.

The utility menu consist of a number of utilities that users can use on-line. This is a good place to add on third-party or self-programmed modules to a DLG system. Any program can be added to a DLG system if it is able to: a) run in a CLI, and b) break cleanly with CTRL-C.

Language Configurable

DLG is language independent. All of the text strings used in the software are contained in a configurable language file. DLG also has the capability of working with alternate Latin based character sets, just as the Amiga itself is capable of working with a number of different Latin based alphabets. DLG is so flexible with language that it is selectable on a user-by-user basis. Character sets and languages are tied to the concept of menu-sets, to provide a simple and consistent way for users to select the language, character set, and menu style of their choice.

DLG Menus

There are two different kinds of DLG menu - there are the "default" menus which present choices in a clear, simple way. Then there are "custom menus" which you, the SysOp, can create yourself, using an ANSI or text editor. You can be as creative as you want with a custom menu, adding special colourful touches to fully customize your DLG system.

Users can choose which kind of menu they want from a user-options menu. You, as the SysOp, can create as many different kinds of menus as you wish. A user's choice of menu has other things associated with it: selection of language, and text file set. By selecting a particular menu-set, users are actually changing their entire DLG environment. All prompts, menus, and text displays will be presented in the language associated with the menu set they choose.

Command Stacking

DLG also features a function we call "command stacking". A command stack is simply a series of commands, all typed together at the same prompt. DLG will process these commands as if they all had been typed individually at each prompt along the way. The advantage of command stacking is that it is a much faster way of navigating around a DLG system.

There are two special characters that you can use in a command stack. The first is the semi-colon (;) character, which acts as both a delimiter for some commands, and as a "RETURN" character to use with prompts that have default selections. The other special command stack character is the tilde (~), which acts as an over-ride on the auto-private function when entering the message or file base. Normally, when you enter the message or file base DLG will take you directly to new private mail or private files. The tilde character in a command stack will override this function. This will become more clear later on after you have read about the message bases.

You will get a feel for command stacking as you work through the tutorials. Since you are (for now) unfamiliar with the DLG environment, examples will not be given in this overview.

Hot Keys

For those users who want to dispense with pressing the RETURN key to enter menu choices, DLG has a "hot key" mode. In hot key mode, a menu selection can be made with a single keystroke. Command stacking is still available in hot key mode if you type a semi-colon (;) as the first character. This will switch hot keys off for that one prompt.

Smart Input for SysOps

DLG has incorporated a unique entry system for use when you are entering user names or options in many of the SysOp editors. This is an "intelligent system" which will zero in on a user name or option choice even as you are typing the letters in. As you enter characters, DLG will display the first available name or choice that matches those characters. This usually finds the item you are selecting before you have typed the entire name in. This same input system incorporates a "scrolling" feature, which lets you move up and down the list of choices or names with your cursor keys. This can significantly cut down on the amount of keyboard activity required to run and maintain a BBS. This intelligent input system is used everywhere in DLG where you need to select one choice from a known list of options - be it user names, display options, port names, etc.

Here is how the smart input routine works: cursor left/right will move the cursor to the left and right. Shifted cursor left/right moves to the beginning or end of the string. Cursor up/down will scroll through an array of options (user names, port names, etc.). Shifted cursor up/down will go to the first or last elements of the array. CTRL-X will clear the string. Typing a character will display the first array element that matches the string up to the cursor position. Backspace will back up and display the first array element that matches the string up to the cursor position. Backspacing to the beginning of the string will clear the string.

Input line editing (cursor left/right, shifted left/right, CTRL-X clearing) is also available at every prompt, not just in those prompts that incorporate the smart input mode.

TPT_Handler

All of the DLG system revolves around a central piece of software - the TPT_Handler. It is the handler that opens and closes ports, mediates communication between the ports, and manages the interface between DLG and the rest of your Amiga's operating system.

TPTCron

Bulletin board systems are time dependent. User's are given a certain amount of time for each call. Various maintenance programs need to be run periodically. Mail needs to be packed up and sent to other systems a certain times of the day. Your system might need to change access to adjust for various amounts of traffic during the day, or to provide a window of opportunity for other systems to call yours. You may want to bundle mail for particular users so that it will be ready when they call in. These timing and scheduling needs are provided by TPTCron, DLG's timed event manager. TPTCron is a stand-alone application that can be used to schedule many other tasks on your Amiga system, apart from DLG.

TPTCron functions by reading a list of tasks to perform from a text file called a "CronTab". This file contains a list of tasks to perform, and when to perform them. A task can be set-up to be performed every so many minutes, every so many hours, on a given set of days, and so on. TPTCron can also be controlled by CronEvent, another DLG module. CronEvent can be used to program "one-shot" timed events on the fly. TPTCron also have an ARexx port through which it can be controlled.

TPTCron is the only part of the DLG package which may be freely distributable. An archive for this purpose is already included in the package, and installed for you in your default file area.

TPTRM

TPTRM is DLG's resource manager. With a multi-tasking system like DLG, an overriding manager is required to mediate access to shared resources, such as message areas, file areas, ports, and so on. For example, users reading messages, users writing messages, software exporting messages, software importing messages, mail bundling software, message renumbering utilities, and so on, could all be trying to access the same message area simultaneously. Imagine the chaos that would ensue if such access were allowed. TPTRM controls access to ensure that collisions do not take place. A host of DLG modules, with descriptive names like OpenArea, CloseArea, GetPort, FreeMisc, and so on, communicate with TPTRM.

User-Levels

Each user account has a numeric user-level associated with it. Auto-access message and files also have user-level thresholds associated both with area entry and area access. Individual menu items also have user-levels associated with them. By changing a user's access level, you can change which message and file areas that user can enter, what kind of access the user can have in each message and file area, and what menu commands are visible to that user.

User-levels are used by DLG as simple keys. Most items in the DLG system have two user-levels associated with them - a low user level, and a high user level. Users will only be able to access that item - be it a message or file area, or a menu command, if their own personal user level fits within that item's lower and upper access levels.

User Account Basics

Each user account has a private message and file area attached to it. Users manage their own private messages and files, and there are safe-guards built into DLG to prevent abuse of your hard drive space by users who refuse to manage these things properly. A user account also has a data file which contains all of the various settings and information on that user. The information is divided roughly into four basic types:

- 1) permanent information that neither the SysOp nor the user can alter.
- 2) SysOp definable information that the SysOp can change.
- 3) user definable information that a user or the SysOp can change.
- 4) system information that changes with each new call.

Here are some examples: Permanent information would include the user's name, birthdate, etc. SysOp definable information would include the user's upload/download ratio, NetMail credits, and so on. User definable information would include the password, areas to search, etc. System information would include time of last call, files downloaded, files uploaded, and so forth.

A user is able to specify a series of message and file areas to check for new messages. These are called "global new scan" areas. The "N" newscan commands in the message and file areas depend upon this being correctly set when a user attempts to perform a newscan. DLG is set up so that when new users log onto your system, all of the message and file areas that they have access to will be added automatically to their newscan areas. DLG will also ask you if you want new areas that you create to be added to the newscan lists of existing users. You can also add newscan areas when you edit user's user levels. In this way, the newscan feature will work largely without user intervention. They will likely only want to use it to turn off areas they are not interested in scanning.

Users are also able to specify a particular file transfer protocol (e.g. Zmodem) for uploading and downloading files. If users do not choose a default, then they will have to choose each time they want to transfer a file.

Other user-definable options include settings for screen size, colour, ANSI settings for colour and positioning, choice of editor, and so on. With the default DLG user application form, users choose these options as they are applying for membership on your system.

Message and File Area Basics

Message and file areas in DLG are very similar. They each feature similar menus and facilities for reading new messages and files as they appear in each area. In the case of file areas, new files are presented as messages - the message being the description for the file. This presents to the user a consistent and easy to grasp way of working with files and messages.

There are two kinds of message and file areas. There are auto-access areas, which are "keyed" by user-levels, as discussed above. There are also special access areas, which have restricted memberships. Users will be added to, or removed from, auto-access areas as their user-levels are changed by you. Users must be manually added to, and removed from, special access areas by you. Generally speaking, it is more convenient to use auto-access areas for most, if not all, of your area set-ups, and reserve special access areas for those situations that cannot be adequately resolved through user-levels alone.

DLG Startup and Quick Tour

In this chapter we will start up your DLG system and take you on a quick tour of the software to familiarize you with its structure and layout. Please work with the software at your computer while reading through this chapter. **NOTE:** *It is important that you take this tour, as you will be making important configuration changes to your DLG system at the same time!*

General preparations:

Once you start up the DLG software, it will be ready to answer any calls that come in on your modem. We suggest that before you complete your DLG configuration that you either turn your modem off or take the phone off the hook. Once you start DLG, we will discuss ways to turn the software off again.

Amiga CLI

Many aspects of running DLG depend upon you entering commands in a CLI. If your Amiga closes the CLI upon startup (as the default Startup-Sequence does), you will probably want to make a modification to your S:startup-sequence file (Workbench 1.3) or S:User-Startup (Workbench 2.x) to open a Shell each time you start your machine. If you do not make such a modification, you will have to open a Shell from the Workbench each time you want to access DLG.

Starting DLG

The installation utility put several script files into your S: directory. A script file is a series of CLI commands that instruct your computer to perform a series of tasks. For most examples in this, and other chapters, to work, you will need to have a PATH to your S: directory. A PATH tells the computer where to look for commands and script files. The default startup-sequence files for Workbench 1.3 and Workbench 2.x include a path to S:.

The installation program put a script file into your S: directory called "DLG.Start." This script file contains all the commands needed to start DLG each time you power up your computer. To start DLG manually, you need to type this into your CLI:

```
DLG.Start
```

Once DLG has started, it will be running in the background, awaiting a caller. If you would like to have DLG started each time you turn your computer system on, then we suggest that you edit your S:Startup-Sequence (for Workbench 1.3) or your S:User-Startup file (for Workbench 2) to include the following line:

```
execute S:DLG.Start
```

If you elected to install the FidoNet utilities from Disk3, you will see some messages printed on your screen about configuration files that need to be set up. We suggest that you ignore these messages for the time being, and concentrate on configuring your BBS before you begin to work with the FidoNet software.

Logging in to DLG

To log into DLG, type the following into your CLI:

```
local
```

This will execute another batch file that the install utility placed in your S: directory and will initiate a local log-in session.

When you installed the DLG system on your hard drive, the installation program asked you for a name and a password. The installation program used that information to create a SysOp account for you. A SysOp account has the highest level of access on your DLG system, and will enable access to the SysOp utilities, where you can control and configure your system. You will need the name and password to be able to log into your system, and configure it for the first time.

DLG will prompt you for your name and password. Use the name and password that you gave during the installation process. DLG will display various messages, and then print out a menu of commands. This first menu is the MAIN MENU.

```
TL0: JAMES HASTINGS-T B:19200 ON:21:11 TA:1439 T:Aniga
Process 6: Loaded as command: Status
System Was Last Re-Booted on:
Friday 10-Jul-92 21:09:18
Checking for new public messages addressed to you...none found

No new bulletins...

Main Menu
-----
[M] Message Sub-System      [F] File Sub-System
[P] People-Talk Sub-System  [C] Chat With Sysop
[U] User Options Sub-System [B] Bundle mail for offline reading
[U] Utility Sub-Section     [S] Sysop Sub-System
[D] Direct Mail to Sysop    [G] Goodbye - Log Out
[?] Display menu           [H] Help
-----

Select => █
```

A Quick Tour of the DLG System

The following guide will take you on a quick tour of the DLG system. This will get you used to navigating around your system quickly. The tour is set-up as a series of commands. You will see a plain English description of what action you should take, followed by the keystrokes (in parentheses) needed to perform that action. (Remember, you can redefine *all* of these commands to any key you desire!) For example, you might see this instruction:

Enter the File base (F)

In this example, to enter the file base, you would type the letter F and then press RETURN to enter the command.

In many cases, you will see a series of keystrokes. These are "command stacks" and can either be typed in their entirety, or entered one at a time. The exception would be where you see the semi-colon character repeated twice or more in a command stack (;). As discussed in the last chapter, a semi-colon in a command stack is a delimiter character - it stands for pressing the RETURN key. The first semi-colon encountered in a string of two or more semi-colons is used to delimit, or separate, the preceeding commands from the rest of the command stack. Any semi-colons immediately following the first semi-colon are then taken to mean the same thing as pressing RETURN at each following prompt. For example, you might see the following:

Get a list of available message areas (a;;)

This would mean the same as typing "a" and pressing RETURN to select the "areas" command, and then pressing RETURN a second time to get the listing. If you want to put two returns following a command, you put three semi-colons (;;;) in the command stack, and so on.

When you log-in to your DLG system, you come to the Main Menu. All of the other parts of DLG branch off from this main menu.

Enter the user-options editor (O)

```

TL0: JAMES HASTINGS-T B:19200 ON:21:30 TA:1423 T:Aniga
[ 1] Name      - James Hastings-Trew  [ 1] Alias      -
[ 2] Address   -                      [ 3] Terminal   - Aniga
[ 4] City      -                      [ 5] Password   - *HIDDEN*
[ 6] State/Prv -                      [--] Joined    - Fri 9 Jul 92 21:06
[ 7] Country   -                      [--] Lst Login  - Fri 9 Jul 92 21:20
[ 8] Pst/Zip    -                      [ 9] Phone No.  -

[10] Help Level - Novice  [--] User Level - 255  [11] Screen Len - 23
[12] More Prmpt - No     [--] Dly Limit   - 1440  [13] Scrn Width - 80
[14] Ansi Flag  - ----  [--] Sesn Limit - 1440  [15] Msgs Read  - 0
[15] Hot Keys   - No     [--] Used Today  - 16   [16] Files Upld - 0
[16] Lex Flag   - No     [--] Num Calls   - 3    [17] 'K' Upld   - 0
[17] Pop Screen - No     [--] Dir Limit   - 10000 [18] Files Dnld - 0
[18] UUCP Priv  - NONE   [--] Msgs Wrtn  - 0    [19] 'K' Dnld   - 0
[19] In Online  - 14     [17] Lst MArea  - 999  [20] Sysop Pgs - 0
[20] Bltn Write - Yes    [18] Lst FArea  - 1    [21] U/D Ratio  - NONE
[21] Credit     - 10000  [19] Protocol  - D:- U:- [22] Capture   - NONE
[22] Editor     - 0     [20] NetMail    - WS   [23] Global Msg - ...
[23] Global File- ...   [21] Global Arc  - ...  [24] Archiver   - 0
[24] Menu Set   - 0     [22] Char Set   - 0
[25] Login Stack -

Number to edit [RETURN to exit] => █

```

This panel is where users can customize their own user accounts. Some of the things that they can change are the type of editor available, upload and download protocols, new scan areas, password, and so on. Any item here that is marked with a number can be altered. Any item that has dashes [--] next to it cannot be altered by the user. To alter an item, all you need to do is to enter the number and you will either toggle the item, be presented with a small menu of options, or enter a mini-editor designed to allow modification of the item.

Choose a new editor (21)

You will see a short list of text editors available to you.

Select the SysOp editor (4)

Return to the main menu (press RETURN)

Enter the conferencing software (P)

Create a new room (C)

You will be asked for the name of the room (Tutorial)

You will be asked for a number of other items. **Just press RETURN** at each prompt to select the default.

Once you have created a conference room, you will notice that you have a "split-screen" - what you type is shown at the bottom of the screen, but not entered into the conference until you press RETURN. Type something and press return. See what happens?

Now, leave the conference room (exit)

Go to the main menu, and then to the message base (MM)

This is the DLG message base. Here you and your users can exchange messages. Each time you enter the DLG message base, you are taken to the last area that you visited. Since this is your very first time on your DLG system, it has taken you to your private mail directory.

Choose a new message area (a)

Get a list of available areas (press RETURN)

Choose area 1 (1)

You are now in one of the default DLG message areas. These were created for you by the installation program. You can change the name or even delete the defaults if you want. They are provided only as a starting point.

Have a look at the message menu, and see the various options available. Note that you will not see all possible commands at all times. For example, you will not see "REPLY" on the menu until you have actually read a message.

Now, return to the Main Menu (M)

Enter the File base (F)

DLG has taken you to your private file area. In the same fashion as with the message areas, DLG will take you to the last area you visited. Since this is your first visit, DLG has taken you to your private file area.

Get a listing of available file areas (a;;)

Choose area 1 (1)

Note that DLG is telling you how many files are in the area, and what file you are currently at.

View the first file description (press RETURN)

You will see a message describing a file that the installation utility placed in your default file area. This file is "TPTCron" and is a part of the DLG package which can be freely distributed. This default area was installed as a starting point for you. You may rename or delete it if you wish.

Now, lets visit the Utility Menu (MU)

Here you see a menu with various utility programs available on it.

Choose Today In History (T)

You will see an event in history which occurred on this calendar date.

Choose Drop To DOS (D)

You will now find yourself in a CLI environment. This CLI differs slightly from your standard Amiga CLI in that you will be unable to directly execute script files. The Drop To DOS function has safeguards built in so that users who are given access to the command cannot do any damage to your system. As a SysOp, you have pretty much the same power that you do from your normal CLI, so be careful! You can also define exactly what kind of access that users or Co-SysOp's can have in the Drop To DOS mode, or even set up your menu so that this option is invisible to all but yourself.

Get a directory (dir)

When you drop to DOS, your default directory is your user directory.

type a file (type user.data opt h)

CD to the root directory (cd :)

Get a directory (dir)

CD back to your user directory (cd user:your_name)

(note that the name has underscore characters in place of spaces. If your name was "Robin Novak", you would type cd user:robin_novak)

Return to the Utility Menu (exit)

Now, lets visit the SysOp Menu (MS)

The SysOp menu consists of a number of different "editors", each of which allows you to alter or configure a different aspect of your BBS system.

The feedback command from the main menu requires the presence of a group account named "SysOp" in order to work (see the previous chapter for an explanation of a group account). When mail is sent to the name "SysOp" you would like it to be re-directed to you. We can do this by creating a group named "SysOp" and adding you as the only member. Let's create that group now.

Select the group editor (A)

Add a new group (A)

Enter the name for the group (SysOp)

Enter the name of the group-op (your name)

Confirm the new group (Y)

Add a member to the group (N)

Enter "SysOp" as the name of the group to add a member to.

Enter your name. Your name should appear on the screen as soon as you type a character or two. This is the "intelligent" name entry routine at work.

At this point, the editor will remain in the name adding mode, allowing you to add many names of a group at one time.

Exit the name adding mode (press RETURN)

Exit the group editor, and return to DLG's Main Menu (MM)

Log out of DLG (GY)

This concludes our mini-tour of the DLG system. You saw many of the menus and options available both to you and your users. You also performed your first act of maintenance by creating the "SysOp" group. This mini-tour has also demonstrated two of DLG's input features - command stacking and intelligent prompts.

The next few chapters are devoted to tutorials in which you will configure your DLG setup to work best with your hardware.

The first step in creating a business plan is to define the business. This involves identifying the products or services you will offer, the target market, and the competitive landscape. A clear definition of the business is essential for developing a realistic financial plan and for attracting investors.

Next, you need to conduct a market analysis. This involves researching the size and growth of the market, the needs and preferences of your target customers, and the strengths and weaknesses of your competitors. A thorough market analysis will help you identify opportunities and threats, and will provide a basis for developing your marketing strategy.

Once you have completed your market analysis, you can begin to develop your financial plan. This involves estimating the costs of your business, projecting your revenues, and determining the break-even point. A realistic financial plan is essential for understanding the financial requirements of your business and for attracting investors.

Finally, you need to develop a marketing strategy. This involves identifying the most effective ways to reach your target customers, developing a budget for your marketing efforts, and implementing your strategy. A well-developed marketing strategy is essential for attracting and retaining customers, and for achieving your business goals.

Required Steps to Configure Your System

Installation of the DLG software, as accomplished in previous chapters, is only part of the process of setting DLG up on your system. There are a number of steps you must take that are required for DLG to work properly. This chapter will guide you through the process. We will cover how to configure your DLG system to work with your particular hardware setup — your serial ports, modems, screen size, etc. DLG comes pre-configured for a basic equipment setup. You will have to change that configuration to suit your particular needs. By the end of this chapter, you will have:

- configured your modem(s)
- configured your screen display(s)
- configured global variables that determine how your system interacts with your users.
- configured your port(s)

As in the previous chapter, you will see a plain English description of the action you are to take, followed (in brackets) by the letter command(s) that you will need to type to make that action take place. When you see a letter in brackets like so: (S), type the letter and then press RETURN.

Logging in to DLG

To log into DLG, type the following into your CLI:

local

DLG will prompt you for your name and password. Use the name and password that you gave during the installation process. DLG will display various messages, and then print out the Main Menu.

SysOp Menu

```
Select => S
Sysop Sub-System
-----
[D] Define / Edit Message Areas      [F] File Area Define / Edit
[E] Edit Message Area Access        [R] Revise File Area Access
[U] User Account Edit               [A] Group Account Edit
[P] Port Configuration              [N] Network Mail Configure
[C] Configure Menus                 [T] Transfer Area Maintenance
[G] General Configuration           [S] SIG creation/edit
[M] Main Menu                       [?] Display menu
[H] Help
-----
Select => █
```

Select the SysOp Menu (S)

The SysOp Menu is a collection of "editors." Each editor allows you to modify a specific part of your DLG system. One editor allows you to create, modify, or delete message areas. Another editor allows you create, modify, or delete user accounts, and so on. Since we are concentrating on setting up your system, we will not stop now to describe all of these editors. We will come back to the SysOp menu later and explore it in detail.

Select the Port Configuration Editor (P)

Port Configuration

To function properly, DLG needs some basic information about how you want your system set up, and what kind of hardware you have. The Port Configuration editor allows you to set up this information. When you select the Port Configuration command from the SysOp Menu, you will see a menu of commands. These commands let you:

- List, Create, Edit, or Delete Global Configuration files
- List, Create, Edit, or Delete Modem Configuration files
- List, Create, Edit, or Delete Screen Configuration files
- List, Create, Edit, or Delete Port Configuration files
- Modify the visibility of Call Log entries based on user level
- List and modify a list of computer types from which the user selects

Select => P

Port Configuration - Main Menu

[P] Port submenu	[D] Displays	[G] Global sets	[O] Modems
[C] Add computers	[L] List computers	[M] Exit	[H] Help
[?] Display menu			

Select => ■

Global Configuration

The first thing you should create is a Global configuration file. This defines the name for your bulletin board and how your board will work with your users. You can have a different global configuration for each port on your board, or you can use the same global configuration on every port in your system. The choice is totally up to you.

Select Global Sets (G)

Select Add Global Set (A)

DLG will begin by asking for a name for this new configuration file. Enter **"New Global Set"** and press **RETURN**.

Once you provide a name, DLG will begin prompting you for information by asking a series of questions. These questions are:

- BBS name
- Pop global screen
- Use custom screen
- Idle timeout
- Verbose pausing
- Open to public
- Login connect delay
- Minimum baud rate
- Default menu
- Character set
- Default command stack
- Forced command stack
- Area to re-route private mail
- Custom Language

Here is a brief explanation of what DLG is expecting at each of these prompts. For now, you can use many of the built-in defaults by pressing RETURN. To see a detailed explanation of each of these items, please see the reference section at the end of this chapter:

BBS Name

DLG will use the name you give here in the menu titles on your board, on the ports that use this Global configuration file.

Pop Global Screen

You have the option of having DLG open a screen each time a caller logs in. If no screen is opened, DLG will run in a "hidden" mode. You can open a screen up at any time, or set up certain user accounts to open up screens when DLG is running in "hidden" mode.

Use Custom Screen

DLG can either open up windows on your Workbench, or on a custom screen of any given resolution / colour depth.

Idle Timeout

This is how much time, in minutes, can elapse without user input before DLG hangs up.

Verbose Pausing

This parameter tells DLG whether or not to print the word "[PAUSED]" when flow-control is used to pause the output of text. (CTRL-S, CTRL-Q).

Open To The Public

A DLG system open to the public can accept new members. One that is closed will only accept logins from members named in a group which you, the SysOp, need to define. Note that this can be done on a port-by-port basis, by creating different Global Configuration files for each port.

Login Connect Delay (seconds)

Some modems need a little time to settle down after a connection with another modem. This parameter lets you specify a number of seconds to wait after a connection is established before attempting to send data out the serial port.

Minimum Baud Rate

The speed you enter here is the slowest baud rate that a caller can use to connect with ports that this global configuration file is used with.

Default Menu

Each of the DLG menus has a name, such as MAIN, FILE, and UTILITY. You can also create and name custom menus. This entry defines the "Main Menu" for the port. For now just enter "MAIN" at this prompt.

Character Set

This would be a default character set that your DLG system will send and receive data with. Users can select their own character set. ISO Latin 1 is the character set for English speaking regions or countries. Normally this should be set to default.

Default Command Stack

A command stack is a series of DLG menu selections. Each user can configure a command stack to be executed at login. The default command stack is given to users as an initial default, but may be edited by the users if they desire. For now, just press RETURN at this prompt, and come back to it later once you are more familiar with your system.

Forced Command Stack

This is the same basic concept as the default command stack, discussed above. The difference is that the user cannot edit or delete the forced command stack — this stack will execute whenever anyone logs into the system. DLG executes the forced command stack before it executes a user's default command stack. Again, just press return at this prompt for now.

Area to ReRoute Private Mail

When users send each other private mail, you have the option of having a copy of those messages forwarded to another message area of your choice. Since we do not have any area set aside for this function right now, press return at this prompt and return to it later.

Custom Language

All of the non-menu text strings that appear in the DLG software are contained in a special language file. DLG defaults to the English.lang file, but you can choose any other available language set. Users will have the option of individually choosing the language set of their choice by selecting an appropriate menu set.

Once you have provided all the information that DLG requests, save the configuration file. You will use it later when you configure your ports.

Modem Configuration

To function properly with your modem, DLG needs information about it. DLG holds this information in a "modem configuration file." DLG comes with several "pre-set" configuration files for various types of modems.

Select Previous Menu (M)

Select Modems (O)

Select List Modems (L)

You will see a listing of all available modem types. If your particular modem is listed, or if you think that one of the other files is a good match for your modem, then skip to the Display Configuration section below. If you feel that your particular modem requires a special configuration file, then continue with the following procedure.

If you have a special or unusual modem, the next thing that you should create is a modem configuration file. You can have a different modem configuration file for each port on your board. For a detailed discussion of the options you will encounter here, please see the Reference section at the end of this chapter.

Select Add Modem (A)

DLG will prompt you for a name for this modem configuration. We suggest that you name it after your particular brand of modem. Once you have provided a name, DLG will prompt you for information. Here is a listing of the information that DLG will prompt you for, followed by an explanation of each item. After each prompt you will see, in brackets, suggested settings that are applicable to most modems. See below for a description of each item:

Init String For a low speed modem, use "ATE0M0X3S0=0."
For a high-speed modem use this init string:
"ATE0M0F1V1X6B0&H1&K1&Y0S0=0S15=8S38=2."

Note: You can configure many high-speed modems by initializing them once and then saving the settings to NRAM (with a terminal program). See your modem documentation for details. A modem with its configuration in NRAM can use an Init String of "ATZ"

Hang-up String (ATH)
Return To Command Mode String (+++)
Reset String (ATZ)
Answer String (ATA)
Ring String (RING)
Lock Mode (Low speed modem "NO," high-speed modem "YES")
Hang-up With DTR ("YES" — see Hang-up with DTR explained below)
BBS Answer Mode (YES)
Maximum Baud Rate (Highest speed of the computer to modem connection)

Init String

This is a sequence of commands DLG will use to configure your modem for each caller. **NOTE:** to put a pause into a modem initialization string, you can use a "w" sequence. The "\" character is used as a sequence escape character in this part of DLG. In order to put the "\" character into your modem strings, you will have include it twice. For example, "\\x" will send the string "x" to your modem.

Hang-up String

This is the command sequence DLG needs to instruct your modem to hang up the phone line.

Return to Command Mode String

This is a sequence of characters that will put your modem into command mode.

Reset String

This is a sequence of characters that DLG need to instruct your modem to reset itself back to a default condition.

Answer String

This is a sequence of characters that DLG needs to instruct your modem to pick up the phone when it detects an incoming call.

Ring String

This is the sequence of characters that your modem will report with when it detects that a call is coming in.

Lock Mode

This option is for high-speed modems only. When the computer and the modem are in "lock mode," they are both set at the highest possible baud rate. Your modem should be configured for RTS/CTS handshaking.

Hang-up with DTR

If you want to ensure that DLG will be able to hang up the modem no matter what, then answer this prompt with "YES." You will have to make sure that the DIP switch settings on your modem are set to "DTR NORMAL."

BBS Answer Mode

If you have initialized your modem to answer the phone (S0=1), then answer this prompt NO. If you have initialized your modem to NOT answer the phone (S0=0) then answer this prompt YES. DLG will then pick-up the phone when it rings.

Maximum Baud Rate

This setting can either be the maximum speed of your modem, in the case of low speed modems, or the baud rate of the computer to modem connection, in the case of a high-speed modem running in lock mode. HST or V.32 modems should be run with a maximum speed of 19200 or 38400 baud. It should be noted that Workbench 1.3 has a maximum baud rate of 19200.

Once you have provided this information, save your modem configuration. You will use it when you configure your ports.

Display Configuration

The Display Configuration determines what DLG will look like when it opens a screen or window on your Amiga. You may have a different display configuration for each port on your board. DLG comes with several pre-set display configuration files which cover a wide variety of situations. If you see a configuration that will fit your needs, then skip to the section below on DLG Ports.

Select Previous Menu (M)

Select Displays (D)

Select List Display (L)

If you need to setup a custom display setting, you can do so following the instructions below. How you set up your display configuration will determine what DLG will look like when you and your users log in.

Select Add Display (A)

DLG will begin by asking for a name for the display configuration file. Once you have provided a name, DLG will begin prompting you for information. Note that here DLG will prompt you for information regarding both a WorkBench window and a custom screen. The option of using either a WorkBench window or a custom screen is set in the global configuration file. Since you can edit that at any time, you need to have both sets of information available here, so that DLG can respond accordingly.

Here is a list of the questions DLG will ask you, with suggested responses in brackets:

X Position Of Def Window	(0)
Y Position Of Def Window	(0)
Width Of Def Window	(640)
Height Of Def Window	(200)
Activate On Open	(Yes)
Font For Window	(topaz.font)
Size Of Font For Window	(8)
Width Of Def Screen	(640)

Height Of Def Screen	(200)
Number Of Bit Planes	(3)
Hi-Res Screen	(Yes)
Interlaced Screen	(No)
Background Open	(Yes)
Font For Screen	(topaz.font)
Size Of Font For Screen	(8)

Here is a brief explanation for each of these items. For a more detailed explanation, please see the Reference section at the end of this Chapter:

*The following parameters affect how a WorkBench window will appear. **NOTE:** Even if you are using a custom screen for your DLG port displays, these settings still affect the Chat window, as this will appear on your Workbench screen.*

X Position Of Def Window

The horizontal position of the upper left corner of a WorkBench screen window.

Y Position Of Def Window

The vertical position of the upper left corner of a WorkBench screen window.

Width Of Def Window

The width in pixels of a WorkBench screen window. Note that the width plus the X-Position combined should not exceed the width of the actual WorkBench Screen.

Height Of Def Window

The height in pixels of a WorkBench screen window. Note that the height plus the Y-Position combined should not exceed the height of the actual WorkBench Screen.

Activate On Open

This determines if the window will become the actively selected window when it opens on the Workbench.

Font For Window

You may use any "non-proportional" font with DLG. Please note that the font name used here is case-sensitive.

Size Of Font For Window

The point size of the font to use on the WorkBench window. This should be of an existing point size. If you choose a nonexistent point size, DLG will substitute the next largest available size.

The following parameters will affect how a custom screen will appear.

Width Of Def Screen

The width in pixels of a custom screen. Usually this value will be either be 320 for a low-res screen, or 640 for a high-res screen, but other values are possible.

Height Of Def Screen

The height in pixels of a custom screen. Usually this will either be 200 for NTSC non-interlaced, 256 for PAL non-interlaced, 400 for NTSC interlaced, or 512 for PAL interlaced screens. Other values are possible.

Number Of Bit-Planes

The number of bit-planes to use for the custom screen. This would be a value from 1 to 4. The number of bit-planes determines the number of colours possible on the custom screen, by the formula $\text{colours} = 2^n$ where n is the number of bit-planes in the display. There is no reason to use more than 3 bit-planes since DLG's local screen will only display 8 colours.

HiRes Screen

Whether to open the custom screen in low-res mode or hi-res mode. Pixels in low-res mode are twice as wide as those in high-res mode.

Interlaced Screen

Whether to open the custom screen in non-interlaced or interlaced mode. There are twice as many scan lines in interlace mode.

Background Open

Determines if DLG should open its screen behind other screens. This is useful in a multi-tasking environment.

Font For Screen

DLG can use any "non-proportional" font for its custom screen display.

Size Of Font For Screen

The same rules apply here as they did for the WorkBench window information. You should select an existing point size for a font. If you choose a nonexistent point size, DLG will select the next largest point size available, and use that.

Once you have provided all the information that DLG requires, save your display configuration. You are now ready to configure your first port.

DLG Ports

A DLG "port" is a means of connecting with the DLG system. DLG by default has two active ports. One communicates with DLG through the keyboard of the system that DLG is running on. This is referred to as the "local" port, and is what you use when logging in from the Amiga's keyboard. The other default port communicates with DLG through the serial port. This is referred to as the "remote" port.

A port configuration file is a collection of all the previous types of configuration files, with some additional information. To create a port configuration file, you choose which global configuration to use, which modem file to use, and which display file to use. This system is very flexible, because it lets you re-use configuration information from each of these different kinds of items.

You will have a separate configuration file for each port that you have on your system. A default DLG setup has two ports — one local, and one remote, so you will have to configure each of them to use your new global, modem, and display configurations.

Each port has a three character name which DLG identifies it by. For example, the local port is called "TL0," and the first remote port is called "TR0." If you are planning on adding additional serial ports to your computer system, you will have to create additional port configuration files for each one of them. We suggest that you name any additional remote ports "TRx," where x is a number to specify the port. For instance, if you added a Supra2400zi internal modem to your computer system, you could run an additional remote port on your DLG system. You would then call this port TR1. Any port with the name in the form "TLx" is considered to be local port. Any other name is considered to be a remote port. For your convenience, seven additional ports, TR1 through TR7, have been pre-configured with the same defaults that TR0 was set to.

Multiple serial port configurations are covered later on in the manual. For now, we will modify the two default ports "TL0" and "TR0".

We are going to be editing existing default port configurations, rather than adding a new port configuration:

Select Previous Menu (M)

Select Port Submenu (P)

Select Edit Port (E)

DLG will list all the available defined port configurations, by their three character names. To choose a port to edit, simply type the three character name at the prompt. To begin, type "TL0" and press return.

```
TL0: JAMES HASTINGS-T B:19200 ON:21:20 TA:1430 T:Aniga
-----
TR7
TR6
TR5
TR4
TR3
TR2
TR1
TR0
TL0
-----
Enter port to edit => TL0
Current port settings
[1] Serial device - console.device
[2] Unit number - 0
[3] Modem file - DEFAULT1200
[4] Global set - DEFAULT
[5] Display file - default
Select item to change [RETURN to save] => █
```

DLG will list the port parameters with numbers next to them. To edit a particular parameter, simply type the number corresponding to the item you wish to change. The parameters include:

- Serial Device
- Unit Number
- Modem File
- Global Set
- Display File

For a full description of each of these parameters, please see the reference section at the end of this Chapter.

Since the TL0 port is the "local" port, the first three items do not apply. You are not connected through a serial device, and you are not using a modem.

If you are satisfied with the default display file, you won't need to edit that either. You will definitely want to change the global configuration file, however.

Change the global set to use the global configuration that you have just created. To do this, type "4" and press return. DLG will list all the available global configuration sets, and prompt you to choose one. Type in the name of the global configuration set that you created earlier.

If you wish to modify the display configuration file, type "5" and DLG will list all the available display configuration files, and prompt you to choose one.

When you have finished, press return, and DLG will save your changes. These changes will take effect the next time you log into the system.

Now, in a similar fashion, edit the TR0 port. Since you are most likely going to be using TR0 on the built-in serial port of your Amiga, you will not need to edit the first two items. You will want to change the "Modem File," "Global Set," and "Display File" settings to use the configurations that you created in the earlier steps. As you select the number that corresponds to the item that you wish to modify, DLG will list the choices available to you and prompt you to choose one of them.

Summary

In this chapter we covered:

- How to configure a global configuration file
- How to configure a modem configuration file
- How to configure a display configuration file
- How to edit a port configuration

Following the same procedures outlined in this chapter, you should continue to create any additional modem, global, display, and port configurations that you might need for your system. If you wish to see your changes take effect, you will need to log out of the DLG system and log in again.

To log out of the DLG system, return to the Port Configure Menu, SysOp Menu, then the Main Menu, and select Goodbye (MMMGY).

If you have changed the default settings for the screens, etc., you will see these take effect once you log in again with the "LOCAL" command.

Example Section

This section contains some examples that will help you set up a DLG system to your personal liking. It takes the form of a question and answer session.

Q "How can I have my DLG board use a different set of colours on my screen?"

A When you create a display file, DLG does not prompt you to provide colour information. DLG ships with display colours based on the standard ANSI colour set of black, yellow, green, red, blue, magenta, cyan, and white. These colours are muted to provide a pleasing appearance on the screen, but some users prefer the standard MS-DCS style colours. When you EDIT a display configuration file, you have the option of modifying the colours used with that display file. An intuition colour requester will pop up on a custom screen, which you can use to edit the colours displayed.

Q "I have an Amiga 3000, and I run my WorkBench in super-high resolution mode. Will DLG use that screen properly?"

A Yes, we designed DLG to be as flexible as possible regarding the screen display. With Super-High resolution mode, you have the possibility of having six DLG screens visible at once on your WorkBench screen. Be sure to select "WorkBench" in your Global Configuration file, and create several different Display Configuration files, with windows located at different positions on the screen, one for each port that you have.

Q "I have a non-Hayes style modem. Will I be able to use that with DLG?"

A Yes, DLG is very flexible when it is talking to a modem. As long as your modem can be controlled through text commands, DLG can talk to it. All the modem commands that DLG uses are fully configurable. You will need to take your modem manual, and carefully consider how you set up your Modem Configuration files. Don't be afraid to experiment.

Q "I plan to have my DLG system interface with FidoNet, using a "front-end" mailer. Is there anything special I should set up in my port configurations?"

A Yes, you will need to look at your Modem Configuration file and set it up so that DLG can use DTR to hang-up the modem. Some modems cannot be hung-up reliably by commands send from "Front-end" software. With the price of long-distance, you cannot afford to have your system accidentally connected with another system all night. You will also need to make sure that your modem has the DTR option enabled. If you have a multiple serial card that does not support DTR, or if dropping DTR hangs up all ports on a single card, then you will not be able to configure your system this way. You will just have to "trust" your modems to behave themselves.

Q "I plan to have my DLG system interface with UseNet, using Matt Dillon's UUCP software package. Is there anything special I need to set up in my configurations?"

A No, there is nothing you need to set up in your port configurations to integrate UUCP functions with DLG. Matt Dillon's excellent UUCP package will do most of the work independently of DLG. A handful of DLG commands are used to interface DLG message areas with those managed by UUCP, and a special kind of user account is used in DLG to accept incoming UUCP sessions. As a general guide-line, you should get your UUCP package set up, and your UseNet connections running smoothly first, independently of DLG. Once everything is going well, you can then do the work to integrate the UseNet areas with your DLG message areas.

Q "I am a visually impaired computer owner, and I have a hard time reading 80 column type on my computer monitor. What should I do?"

A There is a specially designed Display Configuration file that will help visually impaired SysOps see their screens better. It sets up a low-resolution, non-interlaced display, using the Topaz 8 font. If you choose to use that Display Configuration, you will need to modify your own account to make note of the fact that you are using 40 columns for your display.

Q "I am interested in setting up a multi-line system. What steps do I need to take?"

A You will need to create new modem configuration files as necessary, new global configuration files if you intend on having different systems on different ports, and you will have to edit the port configurations for each port that you intend to use. Once you have done that, you will have to edit your S:DLG.Start file so that the additional ports will be activated. If you indicated that you were running a number of ports when you installed DLG, these ports will already have been activated in the DLG.Start file. If you are adding lines beyond the number of ports provided for in the default DLG setup, you will also need to create new mountlist entries in the DEVS:TPTMountlist file, and create new startup files in DLGConfig:batch. You will then need to edit your S:DLG.Start file to mount and activate these new ports.

Q "I have set up a new port, but it does not seem to be working. What do I do to track down the problem?"

A First, check the S:DLG.Start file to make sure that the port is being mounted and activated at startup. Second, check your DEVS:TPTMountlist file to make sure that there is a mountlist entry for your new port. If your are unsure of the format of the mountlist entry, just copy one of the existing entries and edit that to create your new entry. Third, check the port configuration file for your new port, and make sure that you are using the correct serial device and unit number for the port, and that you are using the correct modem setup file for that port. Finally, check in your DLGConfig:Batch directory and make sure that there is a startup file for your new port. If you are unsure of the proper format for a port startup file, copy one of the existing port startup files and edit the copy to create the new startup file. If you create a new mountlist entry for a new port, you will have to either restart your computer, or manually mount and activate the port from your CLI.

Reference

This chapter only lightly covered aspects of the Port Configuration Editor, to help you get your system up and running quickly. The remainder of this chapter contains reference material you might need to refer to when you decide to alter an existing DLG setup. If you are just starting to set up your DLG system, please skip to the next chapter.

Port Configuration Overview

A DLG "port" is a means of connecting into the DLG system. DLG by default has two active ports. One communicates with DLG through the keyboard of the system that DLG is running on. This is referred to as the "local" port. The other default port communicates with DLG through the serial port. This is referred to as the "remote" port.

A port configuration determines things such as which global configuration to use, which modem configuration file to use, which serial port or software serial device driver to talk to, and which display configuration to use for the connection of each port to the DLG system. You have a separate configuration file for each port that you have on your system.

Each port has a three character name which identifies it to the DLG system. For example, the local port is called "TL0," and the first remote port is called "TR0." If you are planning on adding additional serial ports to your computer system, you will have to create additional port configuration files for each one of them. We suggest that you name any additional remote ports "TRx," where x is a number to specify the port. Any port with the name in the form "TLx" will be considered a local port. Any other name is considered to be a remote port. For your convenience, seven additional ports, TR1 through TR7, have been pre-configured with the same defaults that TR0 is set to.

The configuration for a port consists of the name for the port, the name of a modem configurations file, the name of a display configurations file, and the name of a global setting file.

Modem Configurations

There are several defaults that DLG ships with.

We have default modem files for:

- Standard Hayes-command 1200 baud modem
- Standard Hayes-command 2400 baud modem
- U.S. Robotics Courier HST 9600 modem
- U.S. Robotics HST 14400 modem
- Supra 2400zi internal modem.

If your modem falls into one of these categories, we suggest that rather than creating your own modem file from scratch that you try one of the default modem files first.

Here is a list of the information that DLG prompts you for when you create a modem configuration file. Note that the suggested commands listed here are for Hayes-type modems. If you have a different kind of modem, you will need to consult your modem manual to determine the correct information to type in:

Name of New Modem File

To make this easy to remember, use the name of your modem.

Init String

This is the string that will be sent to your modem when DLG first starts up. Generally, the desired effect is to do the following: set the modem into extended verbal result mode — so that it reports "RING" when the phone is ringing, or "CONNECT" if the modem answers the phone and detects a

carrier tone. For Hayes command set type modems, this string will be fairly simple: "ATE0M0X3S0=0." This would set the modem to echo off, no speaker, extended result codes, and no auto answer. For a U.S. ROBOTICS HST 14400 modem, the Init String will be considerably more complex, something like: "ATF1V1X6B0&H1&K1&Y0S0=0S15=8S38=2"

In general, your modem initialization should configure the modem in the following way:

- No echo
- No speaker
- Extended result codes
- No auto-answer

If your modem has extended capabilities, you will want to configure it for the following options:

- No on-line echo
- Verbal result codes
- No data compression (This is greatly subject to the type of modem that you, and your users have. If you have an MNP5 compliant modem, compression can actually slow down the transmission of files, which are usually in a compressed format to begin with. If you are using a V.42bis compliant modem, it will only attempt to compress data if there is some advantage to doing so. Enabling compression on such a modem will speed up the transmission of text-based data, but not affect the transfer rates of compressed files.)
- Smallest modem buffer available (large modem buffers adversely affect DLG features such as pause/resume text, and cancel text, and can adversely affect file transfer times on noisy phone lines)
- RTS/CTS hardware handshake between the computer and the modem

Hang-up String

Normally this would be "ATH"

Return to Command Mode

Normally this would be "++"

Reset String

Normally this would be "ATZ"

Answer String

Normally this would be "ATA"

Ring String

Whatever your modem reports when the phone rings. This is usually "RING" — note that this is case sensitive. You should type EXACTLY what your modem reports when the phone rings, when you have set it to extended verbal result codes.

Lock Mode

This causes DLG to communicate with the modem at the same baud rate (Maximum Baud) regardless of what baud the remote connection has been established at. This is only used with high speed modems. Lock mode automatically causes DLG to use RTS/CTS handshaking. If you experience difficulties with this setting set to "YES," (you can see this if DLG fails to configure your modem when it first starts up) then set this to "NO," and DLG will use a 5 wire hardware protocol, and ignore RTS/CTS conventions. Most modems that operate at 2400 baud and lower will not work with lock mode — you should set this to "NO" if you are using such a modem.

Hang-up with DTR

DLG normally has no problem in hanging up after a user by using the command mode and hang-up codes. If your modem occasionally refuses to return to command mode while it is actively connected to a carrier, then you will have to use this setting. Note that you must have DTR detection set up in your modem dip-switch settings for this to work (consult your modem manual for details on how to do this).

BBS Answer Mode

If this is set to "YES," then your init string should have an "S0=0" in it to turn off the automatic answering feature of your modem. DLG will detect the "RING" message from your modem, and send it a command to answer the call. If your modem refuses to answer the call in this mode, then set this to "NO" and put an "S0=1" into your init string so that your modem will automatically answer the phone without DLG's help.

Maximum Baud Rate

This setting sets the baud rate at which DLG will send commands to your modem. Some modems are unable to connect at a baud rate higher than the baud rate at which they receive the "ATA" command from the computer. For instance, you might have a 2400 baud modem. If you were to configure this setting to 2400 then the modem will be able to connect with callers who are calling at 2400 baud or lower. If your modem does not have this restriction, then even if this is set to 300, your callers will still be able to connect at 2400. When in doubt, set this to the highest baud rate your modem will reliably receive commands at, or your modem's highest rated setting.

Global Configurations

Note that you can have a different global configuration for each port that you use so that, depending on the phone line, your board will behave differently, or, you could use the same configuration file for each port that you run.

Here is a list of the information that DLG prompts you for when you create a global configuration file:

BBS Name

Give your BBS a good distinctive name — one that indicates the kind of board it is likely to be, or what kinds of people you would like to attract. A name like "Zenomorph BBS" will attract science fiction fans, while a name like "The Labyrinth" will attract adventure gamers, etc. A club BBS should have a simple name that indicates what kind of computer user it wants to attract, so a name like "Southern Metropolis Amiga Users BBS" would not be far wrong. A corporate installation would probably require the name of the company or the department that is running the board, so a name such as "PrintWay Service Bureau" would be adequate.

Pop Global Screen

If you select "YES" then each caller to the port that you assign this global configuration file to will cause a special screen or window to appear on your monitor, so that you can view caller activities. If the machine you are running DLG on is also used for other tasks, and you do not want to be annoyed by this automatic screen, then select "NO." You can still active a view screen at any time with a simple CLI command. (See the TScreen and TWindow commands in the chapter on DLG executables.)

Use Custom Screen

When you configure your individual ports, you will give parameters for both custom and WorkBench screens. Which kind of screen that the port will use is determined by this global configuration parameter. If you select a custom screen, then each port will appear on a new screen with the

parameters that you have given to it in the port configuration. If you select a WorkBench screen, then each port will pop up a window on your WorkBench screen at the location and size that you specify for each port. **Chat** always uses your Workbench settings.

Idle Timeout

The "idle timeout" is a time value, in minutes, of how much time can elapse without user input, before DLG decides that the user has wandered off or gone to sleep at the keyboard, and hangs up. The minimum value here is 2 minutes, and the maximum value is 60 minutes. We have found that a value of 4 minutes works well.

Verbose Pausing

If you select "YES" then DLG will print "[PAUSED]" whenever a user has typed a CTRL-S to halt board output. CTRL-Q will erase the "[PAUSED]" notice and resume output. If you select "NO" then DLG will still honor normal XON/XOFF flow control, but no "[PAUSED]" indicator will be printed.

Open to the Public

If you answer "YES" to this question, then callers who do not have a current account on your system will have the opportunity to fill out a membership questionnaire, and gain access to your system — within the limits that you impose on a new user account. If you answer "NO" to this question, then you will be prompted for the name of a group account which will have sole access rights to the port controlled by this configuration. Group accounts are collections of user accounts that can be setup by the SysOp. If you have set a port as a closed port by answering "NO" to this question, then only the members of the group that you name here can log-in and use that port. All other callers will be turned away.

Login Connect Delay

Certain modems need time to "handshake" with other modems. During this handshake period, the modems negotiate several things: if they are compatible with each other, if they support compression, if they support error correction, etc. Without a time delay at this point, DLG would immediately start sending information and prompting for a user's name and password while the handshake was going on. This results in dropped characters and possible problems for the connect. We have found that a time delay of 5 or 6 seconds helps in this situation.

Minimum Baud Rate

This is the slowest BAUD rate that DLG will accept calls at. If you are running a multi-line board, you might want to set one port to have a minimum rate of 300, and another at 2400 or 9600. This way, both lines will not be tied up at the same time with slow callers. Remember, each port can have its own global configuration if you so desire.

Default Menu

This is the menu that DLG first presents to users when they log in. This is normally "MAIN," but you can construct menus of your own with DLG's menu editor. If you would like a custom menu to appear instead of "MAIN," you would supply its name here.

Character Set

All messages are stored on the DLG system in the character set known as ISO Latin 1. Other languages around the world use variations on this character set, to include special characters and accented characters. If you live in a non-English speaking region or country, you may wish to change the character set that DLG uses to send and receive data. All incoming messages are translated to ISO Latin 1 counterparts, and all outgoing messages are translated back to the default character set. With this method, non-English users are giving up some little used characters from the ISO Latin 1 character set to use for the special characters their languages require. Users will have the option of choosing the character set of their choice.

Default Command Stack

DLG supports an advanced form of "command stacking" — the ability to specify a series of commands at a single command prompt. In addition to this, users have a command stack which DLG will execute each time one logs in. You can specify a default stack which DLG will give to each new user as they join your board. Users can edit or delete this stack if they so wish.

Forced Command Stack

If you wish, you can set a default command stack, which DLG will execute before the user's command stack. This will force the automatic execution of various menu commands after the user logs in. For instance, you could set the default command stack to "UTM," and the users would see the "Today In History" listing as they log in each time.

Area to ReRoute Private Mail

As a SysOp, you may wish to monitor private communications between the members of your board. Reasons for wishing to do so vary from SysOp to SysOp, but most find that they need to be sure that their board is not being used for illegal activities. If you do choose to monitor private messages, you can set up a special message area, with yourself as the sole member, and have all private mail entered on the system duplicated in that area.

Custom Language

All of the non-menu text strings that appear in the DLG software are contained in a special Language file. DLG defaults to the English.lang file, but you can choose any other available language set. Users will also have the option of individually choosing the language set of their choice by selecting an appropriate menu set.

Display File

DLG requires some information on the type of screen that you would like each port to run on. Some users run on NTSC systems (maximum normal resolution of 640 * 400), some users run on PAL systems (maximum normal resolution of 640 * 512), and some are using mega-pixel display systems using special monitor hardware and software. You can configure DLG's screens or windows to conform to how you want them to appear on your system.

When you create or edit a Display file, you will be giving parameters for both kinds of display — custom screen and Workbench window. DLG determines which one gets used by the "Use Custom Screen" setting in your global configuration file.

Here is a listing of the parameters DLG will ask you to supply when you create a display configuration file:

Name Of New Display

The name by which DLG will know this display file

X Position Of Def Window

This is the horizontal position of the upper-left corner of a WorkBench window, measured in pixels from the left-hand edge of the screen. Entering a value of 0 here will position the window directly along the left-hand edge of the WorkBench screen.

Y Position Of Def Window

This is the vertical position of the upper-left corner of a WorkBench window, measured in pixels from the top of the screen. Entering a value of 0 here will position the window directly along the top of the WorkBench screen.

Width Of Def Window

This is the width of the WorkBench window, in pixels. Note that this value will interact with the X-Position that you gave above. The sum of the X-Position and the width should not exceed the width of your WorkBench screen.

Height Of Def Window

This is the height of the WorkBench window, in pixels. Note that this value will interact with the Y-Position that you gave above. The sum of the Y-Position and the height should not exceed the height of your WorkBench screen.

Activate On Open

When DLG opens the WorkBench window, you have the option of having it become the "active" window on your system at the moment it opens, or you can opt to have it open in a deactivated mode. The "active" window is the one that accepts input from the keyboard at any given moment on the Amiga.

Font Name

The name of the font to use on the WorkBench window. This name is case sensitive, and should be in the form "name.font." You should type this font name EXACTLY as it appears in your Fonts: directory. If you wish to use the default system font, its name is "topaz.font." You may only use non-proportional fonts with DLG — proportional fonts will work when displaying messages and menus, but will not work correctly with prompts and the text editors.

Point Size

You may use any size of font with DLG. Some fonts, such as the system font "Topaz," come in a variety of font sizes. Some sizes are provided for non-interlaced screens, while other font sizes are provided for interlaced screens. The size of the font that you choose will affect how many characters will be visible on the screen at once, vertically and horizontally. If you enter a nonexistent font size, DLG will choose the next largest existing font size.

Width Of Def Screen

This is a width, in pixels, for the size of the custom screen. If you are using a hi-res screen, enter 640 here, and if you are using a low-res screen, enter 320. You may enter any value you like here — DLG does not perform "reality" testing, but instead strives to be as compatible with existing and future hardware revisions of the Amiga as possible.

Height Of Def Screen

This is a height, in pixels, for the size of the custom screen. See the chart below for possible values:

	Non-Interlaced	Interlaced
NTSC (North American) Amiga	200	400
PAL (European) Amiga	256	512

Again, DLG performs no reality testing on the values you enter here. If you tell DLG that you want a 320 by 200, interlaced, hi-res screen, then it will do its best to accommodate you!

Number Of Bit-Planes

The number of "bit-planes" directly determines how many colours are possible on that screen. A one bit-plane screen shows two colours — foreground and background. A two bit-plane screen can show four colours, a three bit-plane screen can show eight colours, and a four bit-plane screen can show sixteen colours. As you add bit-planes to an Amiga display, the slower it will scroll — it will even start to adversely affect system performance. Also, a hi-resolution, interlaced screen with three

bit-planes will eat up 96K of your Chip RAM, while a hi-resolution, non-interlaced screen with one bit-plane will use up only 16K. You must weigh your need for colour and resolution against your need for memory and speed.

Hi-Res Screen

This is the Amiga hardware display mode to use to display the screen type you have set up. A hi-resolution screen can have 640 pixels across, while a low-resolution screen can have 320 pixels across the screen. If you have selected a screen width larger than 320, but choose not to use a hi-res screen, then you will lose screen information off in the "overscan" area to the right of your monitor.

Interlaced Screen

On the Amiga, screens can either be "interlaced" or non-interlaced. Without going into the meaning of this video terminology, interlacing will essentially double the vertical resolution of your screen. This mode can cause a "flickering" effect which seems to bother some people. If you have a North-American Amiga, your normal vertical resolution is 200, and will double to 400 on an interlaced display. If you have a PAL Amiga, your normal vertical resolution is 256 and will double to 512 on an interlaced display. PAL Amigas flicker particularly badly in interlace mode, due to the lower refresh rate of the PAL video standard. If you have set a screen height larger than 200 (256 for PAL) and you choose to use a non-interlaced screen, you will lose information off in the "overscan" area at the bottom of your monitor.

Background Open

This will determine if DLG will open its screen in the background, behind other applications that may be running, or behind your WorkBench screen. If this is set to "NO" then DLG will open its screen in front of any other screens that may be in use. If you work with your Amiga while running DLG in the background, you will want to set this to "YES" to avoid the possible interruption in your work that happens when a user logs in.

Font Name

The name of the font to use on the custom screen. This name is case sensitive, and should be in the form "name.font." You should type the font name EXACTLY as it appears in your Fonts: directory. If you wish to use the default system font, its name is "topaz.font".

Point Size

The point size of the font to use on the custom screen. This should be of an existing point size. If you choose a nonexistent point size, DLG will substitute the next largest available size.

General Display Information

The settings for windows will affect how this display file will bring up a port on your WorkBench screen. You can specify the location and the size of the window on your screen. DLG does no reality checking here — you must supply valid parameters, or DLG will fail to open the window. That is, do not specify an X position and width combination that adds up to more than the width of your WorkBench screen, or a Y position and height combination that is taller than the height of your WorkBench screen.

The settings for screens will affect what kind of custom screen DLG will create for the port that uses this display file. Again, DLG does no reality checking to make sure that rational values are being entered. If you enter 640 and 400 for the screen width and height, and you say NO to interlace, DLG will give you just what you ask for. Since the arrival of the Enhanced Chip Set and third party megapixel displays, there are no guarantees as to what constitutes normal values for screen sizes. Therefore, DLG treats this information that you give it at face value, and will attempt to open whatever kind of screen you ask for, reasonable or not.

For each of these displays, you can specify a font name and point size. If you enter the name of a font that does not exist on your system, DLG will use 8 point Topaz as a default. If you enter a font name that is valid, but an invalid point size, DLG will use the next available point size.

We have included a "DEFAULT" display file which gives you a half-sized window on a non-interlaced NTSC WorkBench, and a normal 640*200 non-interlaced NTSC custom screen. You can alter this file if you do not wish to create your own from scratch. There is also a "DEFAULTPAL" display file for PAL users to reflect their own default screen height of 256 on a non-interlaced screen.

We have included an "IMPAIRED SIGHT" display file for SysOps who have diminished sight. They will find that a 320 * 200 screen provides them with a more readable display, due to the larger letters.

Port Configuration

DLG comes with 9 ports pre-configured with some default settings. You will want to edit as many of these ports as you are using to indicate 5 things:

Serial Device

All the ports, with the exception of TLO:, the local port, default to using Commodore's standard "serial.device" driver. If you are using a third party serial card for your multiple serial ports, you probably installed a special serial.device driver designed to work with that hardware. If this is the case, you would indicate the name of the third party driver here, instead of Commodore's.

Unit Number

This is the hardware serial port unit number that this DLG port should talk to. The built-in serial port of the Amiga is unit 0.

In the case of the Commodore's A2232 multiple serial card, each port uses "serial.device" but is assigned a different unit number to differentiate it from the others. The built-in serial port on the Amiga is serial.device, unit 1 and each additional port gets a higher unit number. If you wish to refer to the built-in serial port as unit 0, you can do so by selecting it with the special version of preferences that comes with the A2232. If you are using multiple Supra 2400zi modems, the modem in the first slot is modem0.device, unit 0, and each additional Supra modem has a higher unit number.

If you are in doubt about what number to use here, please consult the manuals that came with your serial card to see how the individual ports are to be addressed.

Modem File

This is the name of one of the supplied DLG modem configuration files or one which you created yourself.

Display File

This is the name of one of the supplied DLG display configuration files, or one that you set up yourself. This will determine the parameters for a custom screen, and a WorkBench window for this port. DLG determines which one to use by your global configuration file.

Global Set

This is the name of a global configuration file which you set up.

General Port Configuration Information

You will have to edit each of the ports to make sure that each port is using your chosen modem file, display file and your global configuration file.

We have defined all the pre-defined remote ports (TR0-TR7) in exactly the same way — DEFAULT modem file, DEFAULT display file, DEFAULT global configuration file. The only difference is that they are assigned to different unit numbers for the serial.device.

Here is a list of settings for port TL0:

Current Port Settings

```
[1] Serial Device - console.device
[2] Unit Number   - 0
[3] Modem File    - default1200
[4] Global Set    - DEFAULT
[5] Display File  - DEFAULT
```

Here is a list of settings for port TR0:

Current Port Settings

```
[1] Serial Device - serial.device
[2] Unit Number   - 0
[3] Modem File    - default1200
[4] Global Set    - DEFAULT
[5] Display File  - DEFAULT
```

For the additional remote ports, you will likely need to change either the Serial Device, or the Unit number or both. If you are using Commodore's A2232 multiple serial card, then all of your other ports will also use the same serial.device, but you will need to change the unit number for each port. If you are using Supra 2400zi internal modems, you will need to change the serial device to modem0.device, and you will need to enter a different unit number for each of your modems. Modem0.device, unit 0 would be the first Supra modem in the leftmost slot on your machine. The next one would be unit 1, etc. It is not necessary to use the modem1.device, modem2.device, etc. that Supra includes with their modems. We recommend that you use modem0.device and the unit numbers instead, to indicate the different modems.

If you elect to create any NEW ports, bear in mind that you will have to alter some of the DLG files to accommodate those new ports. You will have to create extra entries in DEVS:TPTMountlist, add mount and ActivatePort commands to S:DLG.Start, and create a script file for that port in DLGConfig:Batch/[port].startup.

To create a new entry in the TPTMountlist, you need only copy an existing entry in that file, and give it the name of your new port.

To create a new script for that port, copy one of the existing [port].startup scripts, located in DLGConfig:Batch, and edit it to reflect the name of the new port in the commands where it is used.

To modify your S:DLG.Start script file to accommodate your additional ports, open up that file in a text editor and locate the series of MOUNT commands. Add new Mount commands for your additional port names, following the same format as the existing Mount commands. Next, locate the section where there is a series of "DLG:ActivatePort" commands, and following the format that you see there, add new ActivatePort commands for your additional ports.

For your convenience, several Mount commands and several ActivatePort commands are present in the DLG.Start file, but we have commented them out with semicolons. To make the commands active the next time that you execute the DLG.Start file, you need only edit the file, and remove the semicolons from the lines that correspond to the additional ports that you wish to make active.

The Port Configuration Editor

The Port Configuration Editor allows you to create and edit each of the different types of configuration files that are discussed in this chapter. You also have the option of deleting unwanted configuration files with the Port Configuration Editor.

There is one function of the Port Configuration editor that has not been discussed in this chapter. This is the list of computer types. When users log into your system, DLG will ask them about the kind of computer equipment they are using to call the board. They must choose from a list of given computer types. DLG stores the choice that they make as a number in their user account records. For this reason, you cannot edit or delete entries in the computer type list — you may only add to the list of given computer types. If you were to edit or delete an item in the “computer types” list, users would end up with incorrect information in their user account records. Some features of DLG will include batch operations where you can use the computer type in a data-base style query of the user base. Therefore, it is important that this information be correct if you plan on using this capability of DLG.

The First Configuration Editor

The first configuration editor shows you the status and settings of the different parts of your system. You can also use this screen to change the settings of the different parts of your system.

There is one function in the first configuration editor that you will not need to use. This is the "Set to Defaults" button. When you click this button, all the settings of the system will be reset to the default settings. This is useful if you have made changes to the settings and want to return to the default settings. However, you should only use this button if you are sure that you want to reset all the settings. If you are not sure, you should consult the documentation for more information. The "Set to Defaults" button is located in the top right corner of the first configuration editor. It is a small button with the text "Set to Defaults" on it. When you click this button, a dialog box will appear asking you if you want to reset all the settings. If you click "Yes", the settings will be reset to the default settings. If you click "No", the settings will remain as they are.

DLG Message Areas

In this chapter we are going to cover the creation, modification, and deletion of DLG message areas. You will create and configure your first message area and define various aspects that will determine its behavior. After that, you will enter that message area, and compose a test message. Once you have done this, we will cover how to change an existing message area, and how to delete one that you have created. We will also cover how to work with DLG message areas from a user's point of view, and cover some strategies you may want to use when setting up the areas on your system.

Once you have completed this chapter, you will know how to do the following:

- Create a new message area
- Define the characteristics of that area
- Enter a message area, and compose and read messages
- Modify the characteristics of an existing message area
- Delete a message area

DLG Message Areas — An Overview

Message areas are places where you and your users can engage in public discussions on a variety of topics. It has become common in telecommunications to provide a separate message area for each type of topic. You might have a message area for general discussions about the board, an area for SysOp questions and answers, an area to discuss Amiga Graphics, and so on.

Message areas can either be local, or part of a networked mail system. A local message area is one that exists only on your system. A networked mail area is shared by many systems, whether those systems are running on different kinds of machines, within your city, or across the globe.

Message areas can also be either auto-access, or special-access. An auto-access area is one that will allow any user into the area, provided that their user-level fits within the entry level key range. You can assign threshold levels to an auto-access area, by which you control which users can enter that area. A special-access message area is one that will refuse entry to any user who is not a member of that area, regardless of their access. You, as SysOp, have to manually select which users can use a special-access area. If a user does not have access to an area then they will not even see it in area lists.

When you create a message area, you will tell DLG what kind of area it is. It is possible to have a combination of area types. For example, you can have auto-access networked mail areas, or you might want to have a special-access local message area, or vice versa.

In addition to these areas that you create and control, each DLG user has a private message area where they receive private messages. Any private communication that a user receives, either from local or networked mail, will be routed to their own private message area. As a SysOp, you may wish to monitor the private message traffic on your system. DLG will allow you to create a message area for this purpose, and all private mail will be duplicated in that area.

Tutorial — How to Create a Message Area

The kind of message area that we will create in this tutorial is a public message area. Later on, in a different tutorial, we will re-define this area to be a special message area.

To begin this tutorial, you should log into your DLG system. We covered the log in procedure in the last chapter.

Select the SysOp Menu (S)

Select Define/Edit Message Areas (D)

Message area editor			
[A] Add area	[E] Edit area	[L] List areas	[D] Delete area
[M] Exit	[H] Help	[?] Display menu	

Select => █

The Message Area Editor provides you with a number of simple options. These are:

- List Areas. This option will allow you list all the defined message areas on your system. For many operations in the Message Area Editor, you will identify the area to work on by a number, which you assign to the area when you create it. DLG lists the numbers that correspond to the areas next to their names.
- Add Area. This option will allow you to create a message area. DLG will prompt you with a series of questions, and then create the area for you.
- Delete Area. This option will allow you to delete an area that you have created. Doing so will also delete any messages that may be in that area.
- Edit Area. This option will allow you to modify the settings of an existing message area.
- SysOp Menu. This option will allow you to leave the Message Area Editor and return to the SysOp Menu.

Select Add Area (A)

DLG will begin prompting you for information. Here is a listing of the questions that DLG will ask you, with the responses that you should give. We have included a brief explanation of each question. For a more detailed discussion of message area parameters, please see the Reference section at the end of this chapter.

Enter New Message Area Number => 999

This is a number by which you and your users will refer to this area. Message area numbers are also helpful in grouping related areas together. For example, you could have: Amiga-specific areas grouped with numbers like 20, 21, 22, etc.; network mail areas grouped with numbers like 100, 101, 102; and so on.

Please Enter Name Of New Message Area => Tutorial

This name should be descriptive of the kind of messages that are available in the area. This name will appear in the list of message areas, and in the header of message area command menus when you are in that area.

Auto-Access Area [y/N] => Yes

An auto-access area is an area that is "keyed" by user-level. If users do not have a user-level that falls between a specified low and high value, they will not see the area listed in message area lists, and they will be unable to enter the area. If an area is not an auto-access area, it is a special access area.

User Level Required: (lower) => 1 (upper) => 255

DLG will only give you this prompt if you are setting up an auto-access message area. These are threshold values for user access to an auto-access area. In this tutorial, we will set up an auto-access area that new users, with levels from 1 to 255, can enter. We also want to set up the following rules for the area: only users with levels above 10 will be able to write messages in this area; only users with levels above 50 will be able to kill their own messages in this area; and only users between level 250 and 255 will be able to transfer, forward, or kill any message they want in

this area. As a SysOp, you have user-level 255 — the maximum. This setup provides you with the ability to raise some users to "Co-SysOp" status by raising their user-levels, without giving them your full access.

Level required to write: (lower) => 10 (upper) => 255

Level required to kill (lower) => 50 (upper) => 255

Level required to copy/move: (lower) => 250 (upper) => 255

Level required to forward: (lower) => 250 (upper) => 255

Level required to re-edit (lower) => 10 (upper) => 255

Level required for SysOp access: (lower) => 255 (upper) => 255

Your answers to the remaining prompts define what kind of area this message area will be:

Handles Used [y/N] => No

Allow handles to be seen instead of real names on message headers.

Signatures Used [y/N] => Yes

Allow user signatures to be auto-appended to messages when written.

EchoMail Area [y/N] => No

Define the area as an EchoMail area. If you respond "yes," further prompts will ask you for additional parameters for the EchoMail area, such as removing "seen-by" lines when reading, and a custom origin line for messages exported from the area.

NetMail Area [y/N] => No

Define the area as a NetMail area.

UseNet Newsgroup [y/N] => No

Define the area as a UseNet Newsgroup.

Custom Message Translator Batch File => DLGConfig:Batch/

Optional batch file to run to process messages before they are saved to disk.

Message Area Capacity => 100

The total number of messages allowed in this area.

Message Area Renumber Trigger => 500

A threshold value for an external "re-number" utility, to indicate when to renumber this area.

Character Set => (Press RETURN)

Character set to use when saving messages in this area.

Add to users' global newscan areas [Y/n] => No

Each user on your system has a list of message areas that can be tracked by the use of the "N" command in the Message Base. NewScan will search through all of the areas in the list, looking for new messages since the user last logged in. When you create a message area, DLG gives you the option of adding the area to all of the users' newscan lists automatically. If you have a very large user base this can take some time.

Once you have provided all the appropriate information, DLG will create the area for you. If you entered all the information as shown, DLG will have created area 999 — Tutorial. This area allows users between levels 1 and 255 to enter; is not a networked area; uses signatures but not handles; and only allows users with higher levels the ability to write and edit messages.

Entering the DLG Message Base

Exit the Message Base Editor, go the Main Menu, and enter the Message Base (MMM)

Message Area 1

We entered the message base in the quick tour, and the last area we visited was area 1. What you see is a display of information about the area.

```
Select => MMM

Now entering message area [1]: Default Message Area

Total messages -> [0]
Msgs numbered -> [0 to 0]
New messages -> [0]

[E] Enter Msg      [B] Post Bulletin  [O] Edit Signature  [A] Change Area
[S] Change SIG     [P] Private Mail  [C] Reverse Read   [=] Cont Read
[I] Message Filter [J] Thread Toggle [L] Header Scan    [U] List Readers
[^] To file area   [M] Main Menu     [G] Goodbye        [H] Help
[?] Display Menu   [RET] Next Msg    [0] Edit User

Area: [1] [Default Message Area] [0/1-0] => █
```

You can see information on how many messages exist in the area, what the message range for this area is, and if there are any new messages to read since the last time you visited this area.

You will see a menu of commands. Choosing a command will allow you to perform various tasks within the message section of your DLG board. These commands cover the creation and reading of messages, with extra commands designed to help streamline message listing and reading.

We will take a quick break from the tutorial at this moment to present the commands available. This will give you a good idea of how powerful your new DLG system is from the user's perspective. These commands will appear and disappear as DLG determines which should be available to the user at each prompt. We call this a "smart menu" system. Only commands that pertain to the current situation are visible and usable. For example, the "reply" command will not be visible in the menu unless you have just read a message.

Each command is presented with a brief description of its action. For a full description of the Message Base commands, please see the reference section at the end of this chapter:

[E] Enter Msg

Write a message. Prompts on the screen will guide you through the process for creating messages.

[R] Reply To Msg

Reply to the message that you have just read. The reply may be either public or private.

[K] Kill Msg

Delete a message that is either to or from you, if you have just read it.

[B] Post Bulletin

Write a bulletin. A bulletin is a special message that every user who logs into the system will see.

[O] Edit Signature

Create a custom "signature" that will be auto-appended to the end of messages you write.

[C] Correct Msg

Re-edit a message that you have entered on the board.

[A] Change Areas

Move from the current message area to another message area.

[S] Change SIG

Select a SIG (Special Interest Group), which will show you a subset of the available message areas.

[P] Private Mail

Go to your private mail area.

[D] Delete All

When you are in your private mail area, this command will allow you to delete all of your private messages at once.

[N] Next Area

Go to the next message area in your "global newscan" list that contains new messages.

[L] Lex Check Msg

Perform a "readability" test on a message.

[F] Forward Msg

Copy, move, or forward a message you have just read to another message area, or another user, under the original author's name, or under your own name.

[>] Forward Read

Read messages in a forward chronological order.

[<] Reverse Read

Read messages in a reverse chronological order

[=] Cont Read

Read messages in a continuous stream, with no pause or prompt between messages. You will have the options of turning off ANSI colour and MORE prompts.

[^] To File

Jump to the file base. This same command in the file base will bring you to the message base. This is provided as a quick means of going from one part of DLG to the other without having to go through the Main Menu.

[I] Filter

Activate a filter so that you will read only messages that contain the filter string in the To:, From:, and Subject: fields of a message.

[J] Thread On/Off

Toggle "thread" reading on and off. This allows you to read a series of related messages grouped by subject, one thread at a time, rather than reading the messages interspersed with other subjects in chronological order.

[Z] Skip Thread

Skip reading the current "thread" when in Thread Reading mode.

[.] Header Scan

The "Header Scan" command will allow you browse through the messages in a particular area, reading only the To:, From:, Date:, and Subject: fields of a message header. Header scan mode brings up a separate list of commands that allow you to:

[R] Display Msg — display the contents of the current message

[.] Tag Msg — add this message to a list of messages to read later

[Z] Skip Thread — skips showing headers from current thread

[T] Tag Thread — add the entire thread of messages to read later

[A] Abort — end header scan mode

[RET] Next Msg — show header of next message

[?] Disp List — display list of currently available commands

[+] Read Reply

Allows you to read the reply to the current message, if one exists.

[-] Read Original

Allows you to read the message the current message is a reply to, if that is the case.

[T] Tag Read

Read messages that have been tagged by you, or by DLG when messages were addressed to you.

[#] Set High Message Pointer

Resets your high message pointer to the current message. This is a hidden menu item.

[U] List Readers

Show all users who are active readers of the current message area. This command is unavailable in alias message areas.

[M] Exit

Return to the Main Menu.

[G] Goodbye

Log-off the DLG system, ending your current session.

[H] Help

Read help on a command, or general help on the DLG message base.

[!] Re-Read current

Re-read current message.

[?] Display Menu

Show the message base menu.

Tutorial - Entering a Message

In the following tutorial we will enter a message, use some of the editing features of DLG's full screen editor, and save that message.

Go to the message area you created in the previous tutorial (A999)

Once you are in area 999 "Tutorial," you will find that it contains no messages. You are going to compose a new message at this point to familiarize yourself with the process.

Select Enter Message (E)

The message will be not be private (N)

Enter who the message is addressed to (All)

Enter a title for the message (Title)

At this point the display will change. This is DLG's full screen editor. You will see two lines at the top of the screen. One of them shows you who the message is from (yourself) and who the message is to (All). The second line shows you the title of the message. In the middle of the screen you will see the editing area. At the bottom of the screen you will see a status line and an instruction line. The status line shows you that you are composing a public message, and that you are in "insert" mode. The instruction line shows you some very basic commands on how to save your message, abort the creation process, or how to get further help.

Press your ESC key once, and type "?" (Note: pressing the **HELP** key on your keyboard provides the same function)

The editing space is now filled with a set of help commands for using this editor. You can switch from your message and this help screen at any time by simply pressing the ESC key once and typing "?". The help screen shows you all the commands available to you in the full screen editor. The commands available to you are either ESC followed by a character, CTRL key combinations, or sometimes two different ways of activating a single command may be available. This is to make the full-screen editor familiar to a variety of users, who are used to different command combinations to achieve the same results.

Press your ESC key twice to return to editing mode from the help screen.

To compose a message, just start typing. Type the following paragraph:

DLG's full screen editor works just as well on a remote terminal screen as it does on the host computer's monitor. DLG achieves this magic with clever use of ANSI control sequences. In order to use the full screen editor effectively with DLG, a user must have an ANSI compatible terminal. On the Amiga, the full screen editor works with JR-Comm, NComm, AZComm, and any terminal that supports VT-100 emulation.

Now, let's try out some of the editing features of the full-screen editor.

Press CTRL-U

The text cursor will jump from the text of the message up to the TO: field at the top of the screen. Let's change who this message is addressed to.

Press your backspace key three times to delete "All", and then type "DLG Users" and press RETURN. The text cursor will now be positioned after the TITLE: field of the message. Let's change this title.

Press your backspace key five times to delete "Title", and type "DLG's Text Editors" and then press RETURN. The text cursor will now be positioned back into the position it was in the body of the message before you pressed CTRL-U.

We want to insert a new paragraph ahead of the existing one.

Press RETURN twice, and then press your "up arrow" cursor key twice to return to the start of the message. We have now inserted two blank lines at the start of the message. Type the following paragraph:

Welcome to DLG Professional's Message Base. In this section of DLG you will be able to read messages from other users. You will also be able to compose messages of your own. DLG offers two different editors to enable you to do this: A full screen editor, and a line editor.

Now, we want to add a paragraph to the end of the message.

Press your "down arrow" cursor key 8 times to reach the bottom of the screen. Press RETURN once to create a new blank line. Type the following paragraph:

DLG's line editor is for users who do not have an ANSI compatible terminal program. While not as flexible or feature-rich as the full screen editor, the line editor will work with just about any terminal program.

Now we want to edit text within the message. Press your ESC key once, and then type "<" to return to the start of the message. Press and hold your "right-arrow" cursor key until the text cursor is positioned at the start of the word "Professional's". Press and hold your DEL key until the entire word "Professional's" has been deleted. Notice how the editor re-flowed the remaining text while you did this. Now press your "left-arrow" cursor key until you are at the end of the word "DLG", and then type "'s powerful". Notice how the editor pushes the rest of the text back out to accommodate the new text.

This tutorial has only scratched the surface of the commands available in the message editor, but you have seen the most commonly used commands that are available. Let's save this message now.

Type CTRL-Z to save the message. DLG will ask you if you want to save this message. Answer "Y"

```
TL0: JAMES HASTINGS-T B:19200 ON 21:30 IA:1423 I:Amiga
From James Hastings-Trew to DLG Users
Subject: DLG's Text Editors
-----
Welcome to DLG's powerful Message Base. In this section of DLG you will be
able to read message from other users. You will also be able to compose
messages of your own. DLG offers two different editors to enable you to do
this: A full screen editor, and a line editor.

DLG's full screen editor works just as well on a remote terminal screen as it
does on the host computer's monitor. DLG achieves this magic with clever use
of ANSI control sequences. In order to use the full screen editor effectively
with DLG, a user must have an ANSI compatible terminal. On the Amiga, the full
screen editor works with JR-Comm, NComm, AZComm, and any terminal that
supports VT-100 emulation.

DLG's line editor is for users who do not have an ANSI compatible terminal
program. While not as flexible or feature-rich as the full screen editor, the
line editor will work with just about any terminal program.

----- Public ----- Insert -----
Save message? █
```

Once you are back at the Message Area menu, press RETURN, and you will see your first DLG message on the screen. At this time, we suggest that you experiment further with the editor. Enter another message, and play around with all the editor commands to familiarize yourself with them. The better you understand the DLG software, the better you will be equipped to help your users with problems they might experience.

Tutorial - Editing An Existing Message Area

From time to time you will find that you wish to change the accessibility or characteristics of a message area. This short tutorial will show you how to change an auto-access area into a special-access area.

Go to the Main Menu, then the SysOp Menu, and choose the Define/Edit Message Area editor (MSD)

Choose Edit Area (E)

Select the area to edit (999)

```

TL0: JAMES HASTINGS-T B:19200 ON:21:30 TA:1423 T:Aniga
AREA STATS:
[1] Area name: Tutorial AREA: 999
[2] Auto access: YES lower: 1 upper: 255
[3] Access levels:
    write access: lower: 10 upper: 255
    kill access: lower: 50 upper: 255
    forward access: lower: 250 upper: 255
    copy/move access: lower: 250 upper: 255
    re-edit access: lower: 1 upper: 255
    sysop access: lower: 255 upper: 255
[5] Handle area: NO
[6] Signature area: YES
[7] Echo area: NO
[8] Netmail area: NO
[9] UseNet newsgroup: NO
[10] Custom origin:
[11] Message translator:
[12] Message capacity: 100
[13] Renumber trigger: 500
[14] Save char set: 0
Select number to edit or [RETURN] to save: => █

```

Select item 2 to toggle this area from auto-access to special access (2)

Press RETURN and then save the changes.

Recall that a special-access message area is one that users cannot access automatically. If you want a particular user to be able to use that area, you will have to manually add them to that area. We will see how to do just that in the following tour.

User Access to Message areas

To edit user access in message areas, you will need to use a different SysOp editor.

Return to the SysOp Menu, and select the Edit Message Area Access command (ME)

You are now in the Message Area User Editor. Here is a list of things that you can do in this editor, with a brief description of each command. For a full explanation of each option in the Message Area User Editor, please see the reference section at the end of this chapter:

Message base user editor

[A] Add user	[C] Copy users	[D] Delete user	[E] Edit user
[L] List users	[S] List areas	[M] Exit	[H] Help
[?] Display menu			

Select => █

List Area

Lists all areas that a particular user has access to.

[U] List Users

Lists all users who have access to a particular area.

[A] Add User

Add user(s) to a particular message area.

[D] Delete User

Remove a user from a particular message area.

[C] Copy Users

Copy the "membership list" from one message area to another.

[E] Edit User

Edit the access of a user in a particular message area. This access will add to, or subtract from, the default access you defined for the area:

[+] Add [-] Remove [RETURN] No change

Write access:

Kill access:

Forward access:

Copy/Move access:

Re-edit access:

SysOp access:

If you wish at this time, you can now delete message area 999 with the "Delete Area" command from the Message Area Editor (To get there from the Message Area User Editor, type MD). We will not be needing it again.

Now that you have had some experience in creating and using a DLG message area, you should create several message areas at this point. For now, try to create only auto-access areas. When we get to the chapter on "User Accounts" you will be instructed to create a temporary special-access message area. You will be using the skills that you have learned in this chapter.

Conclusion and Tips

You should now know how to create a DLG message area, and have a working knowledge of what some of the characteristics of those areas can be set to. At this point, take some time and plan out what kind of message areas you will want to have on your system. Here are some tips:

1. The majority of your message areas should be auto-access. Control access to those areas by managing user-levels. This is a much simpler method than trying to manage special-access message areas. Reserve special-access areas for those whose membership does not change very often, such as "SysOp Roundtable" and the like. This also allows DLG to perform certain functions like area lists more quickly.

2. Take some time to think about the structure of your message base when you are starting to assign numbers to your message areas. Try to group related areas together by number. Allow yourself room for expansion. For example, you could put one group of areas together starting with 10, another group starting with 20, and so on. If you really want to give yourself room, group the areas by 100s. This makes it easier for you, and your users, to remember where particular areas are likely to be. For instance, all of your Amiga areas could be in the range of 100 to 199. Your Macintosh areas could be in the range of 200 to 299, and so on.
3. Provide enough messages in each area to maintain continuity. For a local message area that will see 2 or 3 new messages each day, a maximum of 100 messages would be adequate to maintain messages, replies, and originals, long enough for a casual user to be able to follow conversations. In an EchoMail area, you would need to consider a larger message area. These areas can get 50 to 100 new messages each day. For the sake of continuity, a maximum of 300 to 500 messages would be better.
4. If you want to monitor your users' private mail, here is what you have to do: (a) Create a special-access message area. Since you are the SysOp (level 255), you will have access to that area no matter what. Set the area to some number that is out of the way, like "999" or even "9999". (b) go to the "Port Configuration" editor in the SysOp menu and edit your "Global Configuration" file. Item 13 is "Private Message ReRoute Area." (c) Select item 13, and enter the number of the area that you just created for yourself. (d) Save this global configuration. Now, whenever a user receives private mail, a copy will be placed in this "ReRoute Area" for you to monitor.

Example Section

Q I want to set my DLG system up as a pay system. When new users apply, I want them only to have access to a few areas. I want the users to have access to different file and message areas based on a scale of fees. How do I do this?

A Create your message and file areas as auto-access areas, with a scale of access levels that matches your planned scale of fees. Create user account templates for the various access levels you wish to assign, and use these templates to adjust user-levels based on the fees that your users pay. They will automatically gain access to the features that they have paid for. If you decide later to demote users because of delinquent fees, then users will automatically lose access to areas as you lower their user-levels. This is much better than trying to create special-access areas and trying to add and delete users individually.

Q I want to be able to designate different users on my system as "area SysOps," to help keep track of areas that I might not have time to look after properly. But, I don't want them to access other areas or the SysOp menu, so I can't do this by adjusting their user-levels. What do I do?

A Set up your message areas as you normally would, as auto-access areas. If the areas in question have existed for some time, then all you need to do is edit the users with the "Edit Message Area Access" editor. For each "area SysOp," you will want to add access in the area they are to look after. You will add these as overrides to the existing defaults in the areas. If you have just created the areas, there will be no users present in the areas to edit. You will have to add your intended "area SysOp" to the area you just created, and then you will be able to edit the access.

Q I need a message area for myself and all of my other "area SysOps" to share ideas and problems. The problem is, they all have various user-levels, and I don't want other users to be able to see what goes on in that area. How do I set this up?

A This is the perfect situation for a special-access area. The membership is not going to be fluctuating, and new members are not going to be added very often. The area has a special function, and the membership is selected across the spectrum of your user base.

Q I want to run my system as an alias or handle system only. How do I set this up?

A set your system up, and make all the message and file areas to be alias areas. When a user replies privately to a message from an alias area, the reply will have the alias shown in the message header when it is read. The potential problems with such a system are:

- (1) When users post private messages from their private message areas, their real names will end up on the messages that are sent. Your users must understand this.
- (2) If you intend on interfacing your DLG setup with FidoNet or UseNet, in most cases aliases are not allowed. You will need to set those areas up as non-alias areas.
- (3) Echomail is exported with real names, even if the area has been set to be an alias area. This is because the messages are stored with the user's real names, and the aliases are only substituted for the real names at the time the message is being read. This is so that users can change their aliases at will. All messages posted by a particular user will immediately show the new alias name chosen, and private replies to public messages will still be addressed to the correct person, even though the alias has changed.

Reference Section

This section contains a detailed discussion of DLG message areas. This information is provided as reference material, and contains some duplication of information already covered in the tutorials in this chapter. If you are just starting to set up your DLG system, please skip ahead to the next chapter on setting up your DLG file areas.

DLG Message Areas

DLG message areas are stored in an assigned volume called MSG: In the MSG: volume you will see a number of directories with numbers as names. When a DLG message area is created, a directory for that area is created in the assigned volume MSG:. The name of the directory is the number of that message area. You will also see a directory with the number 0, which is used to store the bulletins in the system. There will also be directories named "Inbound" and "Outbound" which are used by FidoNet compatible software to place incoming and outgoing message packets. In addition to those directories, you will see a file called "Area.BBS". This file contains a list of all areas on your board, with the default characteristics of each area as you defined them. This file is not human-readable or editable.

A message area directory will contain a number of files. The most numerous of these files will be the messages themselves. These have the name "n.MSG" where "n" is the message number. The other files that you will see are:

- **Pointers.MSG:** This file contains the high and low message pointers for that area. It is used by the RENUMBER program to determine if that area should be renumbered yet (i.e., if it has reached its trigger value). It is also used by the DLG message base to determine how many messages are in the current area. This file is human-readable, and you can edit it with any text editor to change the high and low message pointers, should the need ever arise.
- **User.MSG:** This file contains a list of the users who have access to this area, and their access overrides. It also contains the high message pointers for each user for this message area. This file is not human-readable or editable.

Each message area has parameters that are set when you create the area, and can be edited once the area is created. You can define/edit these parameters in the "Define/Edit Message Areas" command from the SysOp Menu.

One of the attributes of an area is the entry access. Special-access message areas will only allow entry to users who already appear in the User.MSG file for that area. You can add or remove users in this file with the "Edit Message Area Access" editor, in the SysOp menu. Auto-access message areas have a "key-range" — a range of user levels that determine which users can enter the area, and

which ones cannot, based on their own user-level. This key-range can retroactively affect users who are already members of the area. This means that you can raise or lower the entry access key-range, and it will affect user access based on user-level.

Each of the following parameters affects what access a typical user will have in a particular message area. Each parameter here is, again, a "key-range" based on user-level. Any changes you make to these parameters are retroactive — they will affect users who are already members of the area if you alter them. If you need to change the access of a particular user, you will have to use the "Edit Message Area Access" command from the SysOp Menu. Here is a detailed listing of the Message Area Parameters:

Auto-Access Area

An auto-access area is an area that is "keyed" by user-level. If users do not have a user-level that falls between a specified low and high value, they will not see the area listed in the message area list, and they will be unable to enter the area. If you set this to not be an auto-access area, it will be a special-access area. In either case, you will have to answer some questions regarding default user access in the message areas that you define. Once users are in an area, a number of parameters define what kind of access they will have by default. These parameters control things like the ability to enter messages, delete replies, and so on. DLG applies the access that you define here to all users in a message area. As SysOp, you can also modify the access that individual users have in each area. You do this by adding or removing abilities from the default user access, on an individual, user-by-user basis.

The following seven questions allow you to define user access to message areas based on user-level. Lower and upper threshold levels control each attribute, which are "keyed" by users' levels when they are in the area. DLG will prompt you to enter two numbers — a low and a high value for each attribute. DLG will only allow those users whose levels fall between the numbers to have access to the area, or the indicated activity within the area.

User Level Required: (lower and upper values)

These are threshold values for user access to an auto-access area. Only users whose user-levels fall between the lower and upper limits, inclusive, are able to see or enter the area.

Special-Access Area:

A special-access area is one that is closed to all users. If you want a user to have access to a special-access message area, you will have to add that user to that area.

Level required to write: (lower and upper values)

This attribute allows users, whose levels fall between the lower and upper keys, to write messages in an area.

Level required to kill: (lower and upper values)

This attribute allows users, whose levels fall between the lower and upper keys, to delete messages either from them, or to them, in an area.

Level required to copy/move: (lower and upper values)

This attribute allows users, whose levels fall between the lower and upper keys, to copy or move messages either from them, or to them, from one message area to another message area, or privately to another user. A message can only be moved if the user has "kill" access as well.

Level required to forward: (lower and upper values)

This attribute allows users, whose levels fall between the lower and upper keys, to forward messages from this message area to another message area, or privately to another user.

Level required to re-edit: (lower and upper values)

This attribute allows users, whose levels fall between the lower and upper keys, to be able to re-edit messages they have written in an area.

Level required for SysOp access: (lower and upper values)

This attribute allows users, whose levels fall between the lower and upper keys, to be able to kill, correct, transfer, or forward any message, regardless of who it is from or to.

Handles Used (yes or no)

Users choose "handles" or aliases when they apply for membership on your system. When you designate an area as one using handles, the users' handles will be dynamically substituted for their names on the headers of displayed messages. Handles are not exported in EchoMail.

Signatures Used (yes or no)

Users are able to define a special "signature" which DLG will append to the end of each message that they enter. You can suppress this function by turning this attribute off in an area.

EchoMail Area (yes or no)

An EchoMail area is a networked mail area that can be shared by many different systems. DLG treats an EchoMail area slightly differently from a local message area. This is covered in the chapter "Network Mail Overview."

Strip Seen-By Lines (yes or no)

This parameter affects only EchoMail areas. A "seen-by" line contains all the network addresses of the systems that a message has passed through. These can get rather long, and are meaningless to most of your users. This parameter gives you the option of hiding these lines.

Custom Origin Line

This parameter affects only EchoMail areas. Each message entered in an EchoMail area has an origin line attached to it which identifies what software created the message, and the FidoNet address of the originating system. When you set your system up for FidoNet, you will create a default origin line to be used in all of your EchoMail areas. If you wish, as you create or edit each EchoMail area, you may add a custom origin line as an override to the default one that you set up for your whole system. Details on what kind of information you should include in an origin line are covered in the chapter "Network Mail Overview".

NetMail Area (yes or no)

A NetMail message area is one which caters to one-on-one communication between specific users on different systems. When you enter your system into a Networked Mail system, such as FidoNet, the "Net Coordinator" gives you a Node number which identifies your system in the network. You can have one node per telephone line if you wish, but the most common setup is to have one node per system. You may only have one NetMail area on the system.

UseNet NewsGroup (yes or no)

A UseNet area is a networked mail area that you interface to a different type of networked mail system — UseNet. UseNet mail areas are treated very differently by DLG, as they have different needs. We will cover this in the chapter "Network Mail Overview."

Character Set

The Amiga uses the ISO Latin 1 character set internally. Most FidoNet conferences use this same character mapping. For your local and EchoMail conference areas, you should press RETURN to choose ISO Latin 1 (the default) as your area character set. This is totally independent from user character sets and has nothing to do with the availability of accent characters for non-English speaking users of DLG.

If you are in a non-English speaking region or country, you will likely have access to EchoMail conferences that use a different character set as a regional standard. These different character sets allow for use of special accented characters that are either unavailable or not easily used with the ISO Latin 1 character set. In the case where you are creating a message area that will be used in such a regional EchoMail conference, you will need to choose the character set specified for that conference.

This setting also determines what character set DLG will assume that incoming messages from other systems are using. If you have set up an EchoMail area to use the Swedish 7 character set (for example) and a message comes into that area using the PC-8 character set, the message will likely be garbled slightly as DLG attempts to re-map the characters in the message.

Generally speaking, you will likely never need to worry about this setting if you are using DLG in an English speaking region or country, or if you are interfacing your DLG system with English only EchoMail conferences. If you are a non-English speaking owner of DLG, you will likely need to be aware of the standards for the various EchoMail conferences you wish to join as you create areas for them.

Message Area Capacity

This value indicates the maximum number of messages allowed for this area. When the number of messages exceeds this value, DLG will start to delete the oldest messages as the new ones are entered.

Message Area Renumber Trigger

This value indicates that the message area is to be renumbered when the highest message number reaches 500. This will "fill-in" gaps in the message base, and set all the message numbers down to the lowest possible. Renumbering is not automatic — it is an externally triggered event. See the chapter on "How DLG Works" for more information.

Custom Message Translator Batch File

DLG allows for the processing of messages, after they have been written, but before they are saved. This processing can include public domain "translator" utilities which can translate messages written in standard English into a number of humorous colloquial accents and slangs. This translation is performed by running such utilities in standard AmigaDOS script files. Here is a sample of such a batch file:

DLGConfig:Batch/JiveTranslation

```
.key portname/a
.bra "["
.ket "]"
echo ""
echo "Translating Message To 'JIVE'"
DLG:jive < T:[portname].body > T:[portname]temp.jive
delete T:[portname].body
rename T:[portname]temp.jive T:[portname].body
```

When a message is saved, the body of the message is first saved to T: as <portname>.body. The <portname> is the three letter name of the port that the user writing the message is logged in on. DLG then calls the custom message translator batch file and passes to it the port name. In this

example, a supplied utility called "jive" reads in the body file and writes it out in a translated form to a temporary file. The batch file then deletes the original body of the message and renames the temporary file as the body file. DLG will then take the altered body file and use that to construct the DLG message, complete with header and other information, and insert that into the message area.

You can do anything you like in this batch file, as long as you do not require user-input. The batch file must be able to complete its operations without asking for any input from the user.

DLG comes with three public domain programs — Jive, Valspeak, and Kraut. Jive is a translator that converts messages written in standard English into a parody of an American dialect. Valspeak converts standard English into a parody of southern Californian "Valley Girl" talk, and Kraut will add a German accent to standard English text. TelePro Technologies wishes to emphasize that no disrespect to any group is intended by the inclusion of these public domain programs — they are merely used as examples to show the kind of power that DLG allows the SysOp of the bulletin board.

Other Message Area Information

Message Translation

When a message is created, the body of the message is saved to the T: directory, and is given the name "<portname>.body", where <portname> is the three-letter name of the port the creator of the message is currently logged into. This message can be massaged by use of special programs and batch files, mentioned above. It is also subjected to a filtering process, which is normally used to remove inappropriate language from the message.

To create a language filter for your system, use a text editor to create a file called "DLGConfig:Misc/Screen.MSG". This text file should contain pairs of words or phrases, in quotations. See the following example:

```
"fish" "kippers"
"cat" "feline"
"dog" "canine"
```

Every time the string "cat" appears in a message, it will be replaced with the string "feline." DLG will capitalize the first letter of the replacement string if the first letter of the original string is capitalized. If you do not wish the translation to occur in mid-word, you must pad the word with a special delimiter character, "|". In this example, we want the word "cat" to be translated to "feline", but only when it appears as a word, all by itself. The proper way to do this is to enter the translation into the Screen.MSG file as:

```
"|cat|" "|feline|"
```

When the "|" appears at the beginning of a word in the Screen.MSG file, it means that "This is the beginning of the word" and that no other letters should appear before this string of characters. When the "|" appears at the end of a word in the Screen.MSG file, it means that "This is the end of the word" and that no other letters should appear after this string of characters. When the entire string is bracketed by "|" characters, this means that the word is to be taken as an entire word, all by itself.

Once you have set up all of your translation strings, you have to compile the translation file. You can do this by entering the command "CompileScreen" in your CLI. The first step is to "CD" to the directory where your Screen.MSG file is. In the case of the global message screen file, type the following in your CLI:

```
cd dlgconfig:misc
CompileScreen
```

DLG will compile the Screen.MSG file into a file called "Screen.DAT" which allows for very fast message translation. The message filter is also used to filter messages sent by the user using the broadcast module.

Area Message Filters

In the discussion above about message filtering, we talked about the Screen.DAT file that DLG looks for in the DLGConfig\Misc directory. You may also choose to have a different Screen.DAT file in an area for a special effect. For example, in the Humour EchoMail conference area, profanity is usually allowed. If you are filtering profanity elsewhere on your system with a global Screen.DAT file as outlined above, you do not want this to affect users in the Humour Echo. The solution is to create a local Screen.MSG file in the message area directory, and compile it there into a local Screen.DAT file. This local Screen.DAT file will override the global one in DLGConfig\Misc. The local Screen.MSG file will make a simple "substitution" that will leave messages unaffected. For example, let's say that your Humour section is area 100. You would type the following in your CLI:

```
cd MSG:100
ed Screen.MSG
```

(When you are in ED, simply type "a" "a" and save)

```
CompileScreen
```

Your message area 100 will not filter profanity now. What we have done is to instruct the filtering software to substitute the letter "a" for the letter "a" in a message, which causes no change.

Welcome Text

You can have DLG display a text file as a user enters an area. This text file is called "EnterArea.TXT", and is to be placed in the directory of the message area in question. For example, let's say that you want to post a warning in the Humour Echo that there is profanity there for readers who might be sensitive to this. This is what you could do:

```
cd MSG:100
ed EnterArea.TXT
```

(You should use a better editor than ED, but it is free. When you are in ED, type something like the following and save it:)

```
Welcome to the Humour Echo
Sensitive readers are warned that this area contains profane language and adult
content. Proceed at your own risk.
```

Now each time a user enters your message area 100, they will see this message.

DLG will also display a text file as a user enters any EchoMail area. You can either have a global text file called DLGConfig\Text\Echo.txt, or you can have an individual Echo.txt inside the directory of a particular EchoMail message area, as outlined above for EnterArea.TXT. The Echo.txt might consist of a special warning for a particular EchoMail area, or general guidelines for all EchoMail areas, or post the individual rules for each EchoMail area.

User Access to Message Areas

When you create a message area, DLG creates a directory for that area in the MSG: directory. Each message area has its own set of rules — there are ranges that you can set that define the behavior of that area for each user according to user level. Each message area has a list of members, and each area keeps track of what overrides each member has in that area. Each area also keeps track of the high message pointer for each user so that users can easily find new messages in that area.

This creates a flexible environment for SysOp control. You can set up default access values for users in areas, and you will only have to modify users' individual user-levels to make changes to their access. This entails some responsibilities on your part — make sure that you note which areas require what user level, and what attributes require what attribute level, so that when you create user editing templates you can easily find and control what you want to do with each user.

If you need to, you can individually adjust the access given to individual users with the "Edit Message Area Access" editor in the SysOp menu.

Here is a list of things that you can do in this editor:

[L] List Area

After you select this command, DLG will prompt you for the name of a user. Once you provide this, DLG will list all the areas that user has access to.

[U] List Users

After you select this command, DLG will prompt you for the number of a message area. Once you provide this, DLG will list all the users who are members of that area, and will show any special access flags you may have set for them.

[A] Add User

This command will allow you to add users to a message area. If the area is a special message area, you will have to use this command to add the users. If the area is an auto-access area, you will only have to use this command to add those users who would otherwise not get automatic access to the area. DLG will prompt you first for the number of the area, and then for the name of the user to add to that area. The prompt for the user's name will keep coming back until you hit RETURN. This allows you to enter several names with this single command.

[D] Delete User

This command will allow you to delete a user from a message area. DLG will prompt you first for the name of the user to delete, and then will prompt you for the number of the area to delete the user from. Note that if you delete users from an auto-access message area, if the users have sufficient access, they will be able to re-enter that area. To permanently remove a user from an auto-access area it is necessary to adjust the area's or the user's access levels. If you have added users with insufficient access to an auto-access area, this command will allow you to remove them again.

[C] Area Copy

This command will copy the "user base" from one message area to another. Whichever users have access in one area will end up with the same access in another, along with their special-access flags. This is a convenient way of creating several special-access areas that contain similar memberships.

[E] Edit User

This command will allow you to directly edit the access of one user in one message area. When you create an area, you define the *kind* of access that the user can have, based on user-level. For example, you can create a message area that allows users with a user-level of 10 to enter, but requires a user-level of 50 to be able to write messages. By adjusting the area access flags for a user in an area, you can add or remove the automatic user-level based access they got by default. You can set area access flags that will override the user-level ranges for that area. DLG will prompt you for the name of a user. Then DLG will prompt you for the message area to edit that user's access in. DLG will then display the following prompts:

[+] Add [-] Remove [RETURN] No change

Write access:

Kill access:

Forward access:

Copy/Move access:

Re-edit access:

SysOp access:

When you press RETURN at one of these prompts, you are leaving that ability at the default for that area. When you type "+" you are adding that ability to the user, overriding the default based on user-level for the area. When you type "-" you are removing that ability from the user, overriding the default based on user-level for that area.

DLG Message Software

The message areas in DLG are operated through a software module called "DLG:Mess". If you examine the Main Menu (see the chapter on "DLG Menus"), you will find that the "Message Base" entry simply calls the "Mess" module as an executable. You cannot call "Mess" directly in a CLI, as it requires a certain environment to be present so that it can function correctly. However, it is similar to a CLI command, in that it can take certain command line arguments that will affect how it behaves:

```
DLG:Mess [-a <area> -c <command stack> -s <sig> -f <force> -m <menuname> -p  
          <level>]
```

Here is an explanation of each argument:

-a <area> : You can create a custom entry for Mess in your Main Menu that will cause the user to be taken to the given area whenever the message base is entered using that command. Bear in mind though that if the user has private mail, this command line argument will have no effect - the user would enter the message base in the message area you specify, but would be immediately redirected to their private message area. The solution to this is to use the -c <command stack> argument, discussed below.

-c <command stack> : You can create a custom entry for Mess in your Main Menu that will pre-pend a custom command stack onto any existing command stack the user might have. This command stack can then direct the user into a particular course of action. If the command stack starts with the "~" character, then any private mail the user might have in their private message area will be ignored. Otherwise, DLG will redirect the user to their private message area, and forget the rest of the command stack.

You can use the command stack argument to make other arguments work properly. Let's say that you wanted to use the -a <area> argument to take the user to message area 10 from a Main Menu command. The executable parameter for your new menu entry would be:

```
DLG:Mess -a 10 -c ""
```

This would prevent DLG from redirecting the user to their private message area if they had new mail waiting.

-s <signumber> : This command line argument will put a user into a particular SIG when they enter the message area. If you do this the SIG you put them into will NOT become their default SIG unless they change SIGs manually.

-f : This command line argument works with the -s <signumber> argument in that it will force a user to remain in the SIG that you specify with the -s argument. The user will be unable to select a different SIG. This argument works by simply removing the Change SIG menu item.

Using this feature, you can now create several different message area SIGs available from the Main Menu of your BBS. For example, let's say you have created a SIG for Amiga users — SIG number 1. Your Main Menu could have a command "Enter AMIGA Message Base", and you could call Mess with the appropriate command switches when you define the menu item:

```
DLG:Mess -s 1 -f
```

A user entering the message base in this fashion will not be able to switch SIGS without returning to the Main Menu. This feature allows you to create the illusion of having several special message areas. If you leave the -f switch off the command line for Mess then the user will start out in the

indicated SIG as a default, but will be able to change SIGs once they enter the message base. If you call Mess with no switches (as your system is configured as a default), the users are defaulted to the last SIG they visited on their previous call.

-m <menuname>: This is the name of a configurable menu to use instead of the default built-in menu. Using this option can provide you with more than one set of commands for your message areas.

-p <level>: This indicates a private send level. Either the sender or the receiver has to have a userlevel that is at or greater than the level specified here. For example, if you set the -p option to level of 255, then anyone would be able to send messages to the SysOp, and the SysOp would be able to send messages to anyone, but none of the users would be able to send messages to each other.

Message Area Commands

What follows is a detailed explanation of each of the commands available in a default DLG message area setup. If you have configured your message area menu, the commands listed here might not match up with your system. The explanations listed here are the same as those found in the default DLG on-line help system, and are written in the context of a message area USER.

[E] Enter Msg

This command allows you to write a message.

If you are in your private message area, then the messages that you enter are automatically private. If the SysOp has granted you the ability to write NetMail or UseNet messages, then you will be asked if the message is to be local, NetMail, or UseNet.

If you are in a public message area, then you will be asked if you wish the message to be public or private.

Once you have chosen which kind of message it is that you wish to enter, DLG will prompt you for the name of the person you are writing the message to, and for a title for the message. Once you have provided this information, DLG will allow you to enter a message using the editor of your choice. You can choose which editor you prefer in your "User Options," available from the main menu.

There are two types of editor available on the DLG system. Your SysOp may have added other editors beside the ones that come with DLG. One type of editor is the "full screen editor" — this editor allows you to use cursor keys and feels the most natural for creating and editing text. You will need to have an ANSI compatible terminal, or at least VT100 emulation to be able to use DLG's full screen editor. The other type of editor is a "line editor" — this editor is not as flexible and easy to use as the full screen editor, but will work with almost any terminal program.

Prompts on the screen will guide you through the process for creating messages. When you finish creating a message, press CTRL-Z to exit either of the editors. If the line editor was used, DLG will give you the options of: aborting the message; saving it; listing it to the screen; or re-editing it. The full screen editor will give you the option of saving the message, or continuing to edit it.

[R] Reply To Msg

This command allows you to reply to the message that you have just read. If you have not read a message in the current message area, then this command will be unavailable. When you reply to a message you will have the option of making the reply public, or private. If you are replying privately to a message in an EchoMail area, the reply will be sent by way of NetMail, if you have NetMail privileges. However, if the user who wrote the EchoMail message you are replying to has an account on your local system, then the reply will be sent locally, and not sent NetMail. You do not have the option of replying privately to UseNet messages at this time.

When you reply to a message, DLG will address it to the writer of that message, and will also use the title of that message as a default title for your reply. The editor that you use to reply to a message is the same you use to enter messages.

[K] Kill Msg

This command allows you to delete a message that is either to or from you. You have to read a message before you can delete it. Note that a user can only kill a message that is from them or to them, unless they have SysOp access.

[B] Post Bulletin

If the SysOp has granted you the ability to enter a bulletin, then this command will be available to you. A bulletin is a special message that every user who logs into the system will see. A bulletin has a time limit. DLG will display it for a certain number of days before deleting it. Bulletins can also be "one-shot" command stacks, allowing you to direct users to a certain menu item on a one-time only basis.

[O] Edit Signature

A signature is a personalized file that DLG will append to each message that you compose. Some message areas do not allow signatures, but most do. You use the same editor to work with your signatures that you use to enter messages. You can have five different types of signature, for different areas. You can have a Net signature for NetMail, an Echo signature for Echomail areas, a File signature for file areas, a Local signature for local areas, and a UUCP signature for UseNet areas and mail.

[C] Correct Msg

This command allows you to re-edit a message that you have entered on the board. In the case of EchoMail messages that have already been sent to remote systems, you will be informed of this fact, and asked if you wish to correct the message anyway. The corrected message will *NOT* be re-sent. This command will only be visible if you have just read a message that you composed, or if you have SysOp status in the area.

[A] Change Areas

This command allows you move from the current message area to another message area. If you wish to see a list of available areas, then use the "A" command alone, and DLG will give you that option. If you know the number of the area that you wish to change to, then use the "A##" command. Substitute the number of the area you want to visit for the "##."

[S] Change SIG

A SIG is a "Special Interest Group" — a collection of message areas that the SysOp has grouped together under a common theme. A board often has a SIG for Amiga users, a SIG for Macintosh users, a SIG for Computer Graphics, a SIG for game-players, and so on. If you wish to see a list of available SIGs, then use the "S" command alone, and DLG will give you that option. If you know the number of the SIG that you wish to switch to, then use the "S##" command. Substitute the number of the SIG you want to change to for the "##."

When you are outside any SIG, you will be able to list, and visit, all DLG message areas that you have access to. When you enter a SIG, you will only see a sub-set of those areas — just the areas that belong to the SIG you joined. If you wish to visit an area that is not included in the current SIG, you will have to either switch SIGs, or remain outside any SIG. To remain outside of all SIGs and see all available areas, press RETURN at the SIG list.

[P] Private Mail

This command will take you from the current message area and place you in your private message area. Normally when you first enter the message section of DLG, you are taken to your private message area if you have any waiting mail. Also, most DLG message areas will allow you to compose a private message to an individual on the board without having to switch to your private message area. This command is available if you need to re-read some past mail, or if you want to post privately from a DLG message area that does not support private mail, such as a UseNet area. When you enter a private message, DLG will ask you if you want to retain a copy of the message for yourself.

[D] Delete All

This command only appears when you are in your private message area. This command allows you to delete all of your old private messages. You should do this from time to time, as it helps to free up space in your private directory. Your private messages and private files share the same directory space. If you leave too many messages around in your private message area, other users may not be able to upload private files to you, depending on the directory size limit.

[N] Next Area

Under your "User Options" from the Main Menu, you can set up a list of areas to "scan" for new messages each time you call in. When you first joined this system, DLG placed all the areas that were available to you in your "Global Message" list. As you access is upgraded new areas will appear in this list. You may add and subtract from that default list at any time by visiting the "User Options" editor available from the Main Menu. This command will take you from the current area to the next area listed in your "Global Message" list. DLG will scan through each area in your list and only stop at those that have new messages.

TIP: When reading messages in DLG, all you have to do is HIT RETURN. Return will take you to each new message as you read. As you hit the highest message available in an area, hitting return will do the same thing as using the "N" Next Area command, and will take you to the next area in your "Global Message" list that has new messages in it. All you have to remember is to "keep hitting return" and DLG will show you all the mail in all the areas that you are interested in scanning.

[L] Lex Check Msg

DLG can perform a "readability" test on a message. If you have just read a message, this command will be available for you to use. The readability test will indicate a number of things about the message, including what kind of education would be required to understand the message. This information is approximate, and is a feature provided mainly for interest. Note that a user can select Lex checking in the User Options editor. Lex checking will then be performed on every message that the user composes. The Lex Check command here is to allow the user to Lex check messages that other users have written.

[F] Forward Msg

The "Forward Msg" command is really three commands in one. When you have read a message you will have the option of copying it to another message area, or to another user to read privately. If you have the option of killing the message, you will also have the option of moving the message to another area or user. You also have the option of forwarding a copy of the message, under your own name, to another message area, or privately to another user. All of these possible uses of the forward command depend upon what options the SysOp has allowed for you in the current message area. You may also be able to forward messages to NetMail, EchoMail, or UseNet, depending on the access your SysOp has given in your user account.

[>] Forward Read

With this command you can instruct DLG to show messages to you in a forward direction. This is the normal mode for reading DLG messages.

[<] Reverse Read

With this command you can instruct DLG to show messages to you in a reverse direction. Use this command to "back-up" to previous messages. Use the ">" Forward Read command to return to normal reading mode. If you change areas with the "P" Private Mail or "A" Area Change commands, the reading mode automatically switches back to forward reading.

[=] Cont Read

Normally as you read messages, DLG will stop in between each message, and re-display the prompt of the message area. Continuous read mode lets you read messages in one long continuous stream, without pausing between each one. You have a number of options available to you with continuous read mode. You can elect to read either all the areas in your global list of message areas, or you can elect to read continuously all the new messages in just the current area. DLG also offers you the choice of leaving ANSI colour active during the continuous read, and whether to have the MORE prompt active. If you are interested in speed, we suggest that you leave ANSI colour off during continuous read mode, turn off the MORE prompt, and use the normal CTRL-S and CTRL-Q keys to pause and resume reading the messages.

[I] Filter

This command allows you to set a filter, which DLG will use to match with the contents of either the To:, From: or Subject: fields of a message header. You will only see those messages that match the filter. The filter will last until you either turn it off (by again using the "I" Filter command), or you switch to another message area.

[J] Thread On/Off

There are two ways of reading messages in a DLG message base. The first way shows you messages in numerical order. The messages are displayed in the same order that they were entered into the message area. This is "Thread Off" mode. With "Thread On" mode, messages are "linked" together. When a message is replied to, the new message becomes part of the same thread. EchoMail messages are threaded together by subject. This makes it very easy to follow conversations in message areas, because you will be shown all the messages in a single conversation before you are shown the next series of messages. You will see the same messages either way, but "Thread On" mode will show them to you in a more logical order.

This mode can cause some side-effects. While you are in a single area, and you are thread reading, DLG keeps track of the messages you have read so that you do not see them twice. For example, lets say that you have read a thread that goes from message 50 to 51, and then to message 62, and then to message 67. DLG will consider that the highest message that you have read is message 50. DLG will look back to message 51 to start picking up the next thread, but will "remember" that you have already seen that message, because it was part of the original thread. DLG will instead go on to message 52 to start picking up the next thread. However, if you were to read this single thread, and then switch to another message area, DLG will "forget" which messages you have seen. When you re-enter this area, DLG will only "remember" that the last message you read was message 50, and it will start to show you message 51, and repeat the rest of the thread again.

The moral is, if you are using "Thread On" mode to read messages, you should completely read all the messages in a given area before you switch to another area, to avoid this type of message repeat. Also, if you type a message number to read a message directly, or if you reverse the direction of reading messages, this will have the same effect as leaving the area and coming back.

If you find yourself in a thread that you have already read, because of this side-effect, then just use the "Z" Skip Thread command to skip to the next thread of messages to read.

[Z] Skip Thread

If you have selected "Thread On" mode to read messages, this command will become available. This command allows you to skip an entire thread of messages, and go onto the next thread. DLG will treat the unread messages as if you had read them. This is a handy feature if you find yourself reading a conversation that you have no interest in.

[.] Header Scan

The "Header Scan" command will allow you browse through the messages in a particular area, reading only the To:, From:, Date:, and Subject: fields of a message header. Header scan mode brings up a separate list of commands that allow you to:

- [R] Display Msg — display the contents of the current message
- [.] Tag Msg — add this message to a list of messages to read later
- [Z] Skip Thread — skips showing headers from current thread. This will only show up when Thread Reading mode is active.
- [T] Tag Thread — add the entire thread of messages to read later. This command will only appear when Thread Reading mode is active.
- [A] Abort — end header scan mode
- [RET] Next Msg — show header of next message
- [?] Disp List — display list of currently available commands

[+] Read Reply

If the message you are currently reading has a reply, this command will allow you to read that reply.

[-] Read Original

If the message you are currently reading was a reply to an earlier message, this command will allow you to read the original message. This command will not change your current position in the message base.

[T] Tag Read

When you are in header scan mode, you have the option of "tagging" messages to read in full, later. The "T" Tag Read command will show you all the messages that you have tagged in this manner. Also, when you logged into the system, DLG searched for, and tagged, all new messages that were addressed to you. When you log onto the DLG system and you are interested in reading your new mail, you can simply enter the Message Base, and use the "T" Tag Read command to read it.

[#] Set High Message Pointer

This command does not actually appear in the menu. Its purpose is to allow you to re-set your high message pointer to the current message. Normally, as you read messages in an area, DLG will keep track of the highest message that you have read in that area. This is so that DLG will quickly find new messages for you to read on each call. If you want to set your high message pointer to a message that is LOWER than ones that you have actually read, use this command. Think of this command as a kind of bookmark function.

[U] List Readers

This command will show all other users on the DLG system that are active readers of the current message area. This command is not available to regular users in alias message areas.

[M] Exit

This command will return you to the Main Menu, where you can select other DLG sub-systems to visit.

[G] Goodbye

This command will allow you to log-off the DLG system, ending your current session.

[H] Help

This command will give you help on the various commands that are available to you on the DLG system.

[!] Re-Read current

This command will allow you to re-read the current message, but has the advantage of not disturbing your thread read position.

[?] Display Menu

This command will display the full menu, if you are an intermediate or expert user. A novice user will always see the menu displayed on their screen.

Miscellaneous Notes

- Each message in an area has a number which indicates the order that it appears in the message area. Each user in that area has a pointer which indicates the highest message they have read in that area. When the lowest message in an area exceeds the highest message read by a user, that user is removed from membership in that area. This action is performed during message area renumbering, which, as discussed elsewhere, is an externally triggered event. The function of removing old users from message areas only happens with auto-access message areas, so no harm is done. This is merely a way that DLG "housekeeps" to get rid of inactive accounts (see the section on "User Accounts" for other housekeeping functions in DLG). However, if you have set overrides on a user's access in an area, or the user has been manually added, that user will not get deleted by this function.
- Because NetMail involves many systems and SysOps spending time and money, anything that wastes space in a message is generally frowned upon. Therefore, DLG does not append signatures to NetMail messages, even if you have activated that feature in your NetMail area. Signatures can also interfere with many areafix programs.
- DLG allows you to create "group accounts." These accounts are collections of users that you can define with the "Group Account" editor in the SysOp menu. When a private message is addressed to a group, a copy of that message will be forwarded to each member of that group.
- NetMail & UUCP mail that is sent privately to a user or group will be sent directly to their private mail directory. It can also be directly replied to from the private area providing the user has enough credit to do so. When a message is replied to in this fashion the reply is sent to the NetMail area as private mail. The incoming message is also retained in the designated NetMail area.

- EchoMail messages that are replied to privately are sent via private NetMail. This will only be the case if the user the reply is addressed to does not have a local account on your system, and only if the user writing the reply has NetMail privileges. If the user that the reply is addressed to has an account on your system, then the user is given a choice of sending the reply as a local or NetMail message.
- If you have SysOp access in the NetMail area, DLG will allow you to send or Forward NetMail to an unregistered or "private" node. This is a node that does not normally appear in the Nodelist. If you do not have SysOp access, DLG will only allow users with NetMail Write access to send or forward messages to nodes that appear in the Nodelist. Since an unlisted node does not appear in the Nodelist or in the cost accounting portion of MSG:Traplist.CFG (required by TrapList) the cost of a NetMail message to such a node is recorded as \$0.
- When sending a private message, the user has the option to retain a copy of that message in their own private directory. This can be useful when trying to keep track of private conversations.

File Areas

In the last chapter we covered the creation, use, and maintenance of DLG message areas. In this chapter we are going to cover the same topics for DLG file areas. Many of the things you learned about DLG message areas apply to DLG file areas as well. Since there is much over-lap in the ideas and concepts presented, we will assume that you are familiar with the material covered in the last chapter. The overlap material includes;

- What auto-access and special-access mean
- How auto-access areas are keyed by user-level
- How user-levels determine what abilities users get in auto-access areas
- The concept of adding or deleting users from special-access areas
- The concept of positive or negative overrides on individual user access in an area
- How to use the DLG text editor

You will create and configure your first file area, where you will define aspects of the area that will determine its behavior. After that, we will enter that file area and talk about uploading and downloading files from the area. We will then discuss DLG's file area maintenance functions.

Once you have completed this chapter, you will be equipped to do the following:

- Create a new file area
- Define the characteristics of that area
- Enter a file area, and upload a file
- Modify the characteristics of an existing file area
- Edit file descriptions
- Perform file area maintenance
- Delete a file area

As you work through the next tutorial, you will probably notice that it is reminiscent of the tutorial on creating message areas in the previous chapter. This is because DLG treats message and file areas very similarly. You may want to go ahead and use the skills that you learned in the last chapter to try your hand at creating a file area, without following this tutorial. An experimental approach to the software can make the concepts easier to assimilate. Make sure to read and follow the tutorial on "Entering the DLG File Base" later on in this chapter.

Tutorial — Creating a File Area

File areas in DLG are places where you and your users can exchange a variety of files and programs. It has become common in telecommunications to provide a separate file area for each classification or file type. You might have an area for text files, another for Amiga Utilities, another for Amiga Graphics, and so on.

There are three kinds of file area in DLG. The first kind is an auto-access area, and will likely be the most common type on your board. The auto-access file areas function identically to the auto-access message areas. The second kind is a special-access file area. Again, these function identically to the special access message areas. The third type is the private file area. Each user has a private file area where files can be received in a fashion similar to the way in which private mail is handled.

The Philosophy Behind DLG File Areas

File areas in DLG are very similar to message areas. Each file that is uploaded to a DLG file area can have a lengthy description attached to it. This description is like a message from the uploader, to all the potential downloaders. Since the description is the size of a regular message, it gives the uploader a chance to describe fully the file so that other users can decide if they really want the file or not. In fact, the uploaders will use the same editor to enter the file description that they would use to enter messages in DLG message areas, so everything is consistent.

DLG file areas can be read like DLG message areas. When you enter a file area, you can simply press RETURN to read the descriptions any new files in that area. You have a global file scan list of file areas you are interested in. The "N" Next Area command will move you from one file area to another, from your global file area list. You can even "reply" to files, by posting a comment on a file when you read its description.

From the SysOp's perspective, file areas are very similar to message areas. They are created, edited, and deleted in much the same fashion. Most of the same strategies you need to think about when creating message areas also apply to the creation of file areas.

There are differences: Files require extra maintenance that messages do not. They are going to be around on your system for quite some time, and usually take up a fair amount of hard drive space — much more than message areas do. Files need to be moved around, checked for both integrity and potential copyright violations, and so on. DLG gives you the tools you need to perform these functions. You might also be upgrading to DLG from another BBS software package, and have existing file areas. DLG gives you tools to "upload" groups of files so that you can get your new system set up quickly.

Tutorial — Create A File Area

We are going to create two kinds of file areas in this tutorial — an auto-access file area, and a special-access file area.

Log-into your DLG system

Select the SysOp Menu (S)

Select the File Area Define/Edit command (F)

```
File area editor
-----
[A] Add area      [E] Edit area      [L] List areas      [D] Delete area
[H] Exit          [H] Help          [?] Display menu
-----
Select => █
```

DLG will show you a list of commands that are available in the "File Area Editor". Here is a list of those commands, with a description of what each one does:

[L] List Areas

This command will list all the defined file areas on your system. File areas are known by a name and number. You and your users will use the number to select an area to work in, or visit. The name is used to identify the area and the types of files that should be contained there. The listing this command provides shows you both the number, and the name, for each file area on your system.

[A] Add area

This command will allow you to add a new file area to your system. We will cover this process in detail in this tutorial.

[D] Delete area

This command will allow you to remove a file area, and all the files within it, from your system. DLG deletes all the files associated with a file area when you remove it from your system, so use this command with caution.

[E] Edit area

This command will allow you to change the parameters of a file area once it has been created. These parameters include: auto- or special-access, what kind of privileges users will have in the area, and so on.

[M] SysOp menu

This command will exit you out of the File Area Editor and return you to the main SysOp Menu.

Now will will continue the tutorial:

Select Add Area (A)

Enter the number of the new file area (99)

Enter the name of the new file area (Tutorial)

This will be an auto-access area (Y)

DLG will prompt you for user-level ranges for access to this area. These ranges are identical to the user-level ranges described in the last chapter on message areas. Type "1" and press RETURN. Type "255" and press RETURN.

DLG will prompt you for user-level ranges for upload privileges in this area. We are going to demonstrate DLG defaults here. When you typed 1 and 255 as your access range, you provided the File Area Editor with defaults for all the following range-questions. To see how these work, press RETURN twice, and you will see the prompt filled in with these defaults.

DLG will prompt you for user-level ranges for download privileges. Press RETURN twice.

DLG will prompt you for user-level ranges for transfer privileges. Type "255" at the first prompt, and press RETURN twice.

DLG will prompt you for user-level ranges for kill privileges. Press RETURN twice.

DLG will prompt you for user-level ranges for SysOp privileges. Press RETURN twice. Note that at this prompt, DLG defaults to "255" for both entries, because this will be the range most often desired.

DLG will ask you if signatures are to be used in this file area (Y)

The SysOp will need to validate new uploads (Y)

DLG will prompt you for the number of a different file area to re-direct the new uploads to. If you had answered the "validate" prompt "No," then this question would have been skipped. When you specify that uploads to a particular file area are to be validated, uploads to that area are re-directed to a different file area, usually a special-access area, so that you, the SysOp, can validate the files before making them public. Many different file areas can share the same validation area — DLG keeps track of where the files were originally uploaded to. When the file is validated, it will be returned to the original file area.

Enter the number of file redirection area (999)

DLG installed a file redirection area for you, and gave it the number 999.

This will not be a file-requestable area (N)

The Philosophy Behind DLG File Areas

File areas in DLG are very similar to message areas. Each file that is uploaded to a DLG file area can have a lengthy description attached to it. This description is like a message from the uploader, to all the potential downloaders. Since the description is the size of a regular message, it gives the uploader a chance to describe fully the file so that other users can decide if they really want the file or not. In fact, the uploaders will use the same editor to enter the file description that they would use to enter messages in DLG message areas, so everything is consistent.

DLG file areas can be read like DLG message areas. When you enter a file area, you can simply press RETURN to read the descriptions any new files in that area. You have a global file scan list of file areas you are interested in. The "N" Next Area command will move you from one file area to another, from your global file area list. You can even "reply" to files, by posting a comment on a file when you read its description.

From the SysOp's perspective, file areas are very similar to message areas. They are created, edited, and deleted in much the same fashion. Most of the same strategies you need to think about when creating message areas also apply to the creation of file areas.

There are differences: Files require extra maintenance that messages do not. They are going to be around on your system for quite some time, and usually take up a fair amount of hard drive space — much more than message areas do. Files need to be moved around, checked for both integrity and potential copyright violations, and so on. DLG gives you the tools you need to perform these functions. You might also be upgrading to DLG from another BBS software package, and have existing file areas. DLG gives you tools to "upload" groups of files so that you can get your new system set up quickly.

Tutorial — Create A File Area

We are going to create two kinds of file areas in this tutorial — an auto-access file area, and a special-access file area.

Log-into your DLG system

Select the SysOp Menu (S)

Select the File Area Define/Edit command (F)

```
File area editor
-----
[A] Add area      [E] Edit area      [L] List areas      [D] Delete area
[H] Exit          [H] Help          [?] Display menu
-----
Select => █
```

DLG will show you a list of commands that are available in the "File Area Editor". Here is a list of those commands, with a description of what each one does:

[L] List Areas

This command will list all the defined file areas on your system. File areas are known by a name and number. You and your users will use the number to select an area to work in, or visit. The name is used to identify the area and the types of files that should be contained there. The listing this command provides shows you both the number, and the name, for each file area on your system.

[A] Add area

This command will allow you to add a new file area to your system. We will cover this process in detail in this tutorial.

[U] Upload File

When you select the upload command, DLG will ask you if the file is to be sent public or private. If it is sent public, it will appear in the file area that you are currently in. If it is sent privately, DLG will ask you for the name of a user to send the file to.

If you are sending privately to another user, DLG will check and report on how much space that user has in their private directory. If they do not have enough space, the upload will not be successful. The SysOp has no restriction on uploading to a user's private directory, however.

If you do not have a default upload protocol in your user options, then DLG will ask you to select a file-transfer protocol from a list of available protocols. You can have a default file-transfer protocol set up in your user options in one of two ways: (1) When you applied for membership on the board, the application process may have asked you to pick a default transfer protocol for uploads. (2) You can select a default file transfer protocol by visiting the User Options Editor, available from the Main Menu.

The list of protocols that DLG presents you with will show: a letter to choose the protocol by, the proper name of the protocol, a set of flags indicating what that protocol is capable of, and an indication of the relative efficiency of that protocol. Here is a list of what the flags mean:

- D - The protocol is capable of downloading files
- U - The protocol is capable of uploading files
- B - The protocol is capable of "batch" file operations where more than one file can be transferred at once.
- R - The protocol is capable of "resume" operations in case the connection to the board is interrupted.
- F - The protocol requires that you have to specify a filename for the file when you perform the upload.

Generally speaking, the Zmodem file transfer protocol is the best one to choose, if your terminal program supports it. It has the most efficient transfer rates over normal telephone lines, is fairly resistant to file corruption due to line noise, has batch and resume capabilities, and does not require that you enter a filename when you are sending the file.

Whichever file-transfer protocol you select, it has to be one that is compatible with the terminal software that you are using. Once you select a file transfer protocol, DLG will put itself in a mode that is ready to receive a file. If you have selected a protocol that does not send the filename with the file, DLG will prompt you to supply the filename before you can start the file transfer. DLG will then give you a **Continue [Y/n]** prompt before continuing.

You must select the UPLOAD option in your terminal software, and select the file or files that you wish to send.

Once the upload is complete, DLG will present you with the same editor that you use to write messages. You should enter a message describing the file in a fair amount of detail. The first 60 characters or so of the description that you write will be visible in abbreviated listings of the file. If it is possible, try to be as descriptive as possible in the first sentence. You can expand on anything you like in the rest of the message.

If the connection to the board is interrupted during your upload for some reason, DLG will keep the portion of the file that was sent successfully. On your next visit to the File Base, DLG will prompt you to send the rest of the file. This is true only if you have used a protocol like Zmodem that allows for a "resume" for interrupted file-transfers.

Also, when you upload a file, you are given "credit" for the number of bytes in the file. Depending on the setup of your user account, you may have an "upload/download" ratio imposed on your access to files for downloading. Generally speaking, the more bytes you upload, the more bytes you can download. If the SysOp has to delete a file that you upload, the SysOp can decide to subtract the upload credit from your account.

A file request is a FidoNet procedure where a system can call up your system and pass to it a list of files to be sent. DLG, in concert with appropriate "front-end" software, can fulfill these file requests. However, if you do not want, or need, this capability in a particular file area, you can turn it off.

Enter the name of an alternative path to store files: (press RETURN)

Normally, the files in a particular file area are stored in the same directory that the file area itself is contained in. However, if you have extra hard drive volumes or partitions that have lots of room, you can tell DLG to store the files on the alternate volume or partition. Bear in mind that the alternate path can only consist of six characters, including any ending ":" character.

Add to users' global newscan areas: (N)

```
TL0: JAMES HASTINGS-T B:19200 ON:23:55 TA:1389 T:Aniga
Enter new file area number => 99
Name of new file area => Tutorial
Auto access area [Y/n] => Yes
User level required:      (lower) => 1 (upper) => 255
Level required to upload: (lower) => 1 (upper) => 255
Level required to download: (lower) => 1 (upper) => 255
Level required to transfer: (lower) => 255 (upper) => 255
Level required to kill:    (lower) => 1 (upper) => 255
Level required for sysop access: (lower) => 255 (upper) => 255
Signature area [y/N] => No
Sysop validation on new uploads [y/N] => Yes
File area to re-direct unvalidated uploads => 999
File requestable area [y/N] => No
Alternate path to store files [Return for default] =>
Directory created successfully
File area successfully added
Add to users' global newscan areas [Y/n] => █
```

You have now created an auto-access file area.

Entering the DLG File Base

Exit the File Area Editor to the SysOp Menu, then to the Main Menu, and then to the File Base (MMF)

```
Now entering file area [1]: Default File Area.
Total files -> [1]
New files -> [0]

[U] Upload file      [D] Download        [F] File list        [A] Change areas
[S] Change Sig      [N] New scan        [=] Global list      [P] Private area
[T] Transfer file    [O] Edit signature  [R] Read file        [I] View archive
[E] Edit file        [^] Goto msg area   [M] Send message     [M] Main menu
[G] Goodbye          [H] Help            [?] Display menu     [RET] Next file

Area: [1] [Default File Area] [1/1-11] => █
```

When we took the quick tour of DLG in the first tutorial, we visited file area #1. You will see a menu of file area commands. Here is a listing of those commands, with an explanation of each one. These explanations are the same ones that appear in the on-line help:

[F] File List

Normally you simply press RETURN to see new files in any file area — just as you do to see new messages in message areas. Selecting the File List command gives you several options for seeing a list of files, with an abbreviated description. The first four options control what order the files will be listed in:

[F] Natural Order

List files in the same order they were uploaded in.

[B] Inverse Order

List files in order from newest file to oldest file.

[A] Alpha Forward

List files in forward alphabetical order

[R] Alpha Reverse

List files in reverse alphabetical order

Once you have selected the listing order, DLG gives you eight different options to choose which files to list:

[A] All files

List all files in the current file area.

[N] New files

List new files in the current area that you have not viewed. This option will update your high-file pointer.

[S] Last Call

List files in the current area since your last call, including any that you may have viewed. This option will not update your high-file pointer.

[#] Num Days

DLG will prompt you for a number. This number represents the number of days back from the current date. Files in the current area that were uploaded in that time range will be listed.

[C] Since Date

DLG will prompt you for a date in the standard AmigaDOS "DD MMM YY" format — e.g. 01 Jan 91. All files in the current area uploaded since that date will be listed.

[R] Date Range

DLG will prompt you for two dates in the standard AmigaDOS "DD MMM YY" format. All files in the current area uploaded between those dates will be listed.

[F] Filename

DLG will prompt you for a string. All files in the current area that contain that string in their filename will be listed. For example, if you enter "lzh" then all files that have "lzh" anywhere in the filename will be listed. You can also make use of wildcards in your searches. For example, if you enter "~lzh" then all files will be listed except those that end in "lzh". In this case, the tilde (~) character stands for "not". By default, all searches use an implied "*" wildcard at the beginning and end of the search string you specify.

[D] Filename/Desc

DLG will prompt you for a string. All files in the current area that contain that string in either their filename or the first 60 characters of their description will be listed. You can use the same wildcard conventions outlined above for the filename search.

You can use command stacking to automate the file listing process. For example, if you type "FFN" at the file prompt, DLG will display all new files in the current area, in the order that they were uploaded in. DLG will also update your high-file pointer.

[D] Download

When you select the download command, four different things can happen:

- (1) If you have just read the description of a file, DLG will proceed to download that file to you.
- (2) If you have "tagged" files with the "[.] Tag File" file command, DLG will ask if you want those files sent to you.
- (3) If you have tagged files, and then read the description of another file and select the download command, DLG will first ask if you want the tagged files sent. If you answer "No", then DLG will send you just the one file that you were looking at when you selected the download command.
- (4) If you have neither selected a file, nor looked at a file description, DLG will ask you to supply either the number of a file to send, or a filename of a file to send. DLG can only send files that appear in the current file area, unless you have tagged them.

If you do not have a default download protocol in your user options, then DLG will ask you to select a file-transfer protocol from a list of available protocols. You can have a default file-transfer protocol set up in your user options in one of two ways: (1) When you applied for membership on the board, the application process may have asked you to pick a default transfer protocol for downloads. (2) You can select a default file transfer protocol by visiting the User Options Editor, available from the Main Menu.

The list of protocols that DLG presents you with will show: a letter to choose the protocol by, the proper name of the protocol, a set of flags indicating what that protocol is capable of, and an indication of the relative efficiency of that protocol. Here is a list of what the flags mean:

- D - The protocol is capable of downloading files
- U - The protocol is capable of uploading files
- B - The protocol is capable of "batch" file operations where more than one file can be transferred at once.
- R - The protocol is capable of "resume" operations in case the connection to the board is interrupted.
- F - The protocol requires that you have to specify a filename for the file when you perform the download.

Generally speaking, the Zmodem file transfer protocol is the best one to choose, if your terminal program supports it. It has the most efficient transfer rates over normal telephone lines, is fairly resistant to file corruption due to line noise, has batch and resume capabilities, and does not require that you enter a filename when you are receiving the file.

Whichever file transfer protocol you select, it has to be one that is compatible with the terminal software that you are using. Once you select file transfer protocol, DLG will put itself in a mode that is ready to send a file. At this point you must select the DOWNLOAD option in your terminal software. If you have not selected a file transfer protocol that sends the filename with the file, you will have to supply your terminal software with a file name.

If you have tagged a list of files to send, DLG will send all the files to you in one session. If your connection with the board is interrupted for some reason, DLG will remember which files have been successfully downloaded, and which ones haven't. The remaining files will still be tagged on your next visit, including any files that you had partially downloaded. If you are using a file-transfer protocol like Zmodem, you will be able to resume downloading partial files from where you left off.

DLG keeps track of how much data you download from the board. If the SysOp has imposed an "upload/download" ratio on your account, you may not be able to use the download command until you upload something. Generally speaking, the more bytes you upload, the more bytes you can download. A typical upload/download ratio might be 1/10. This means for every byte you upload, you can download 10 bytes — approximately 10 files for every 1 file you upload.

You might see some files flagged with the word "FREE". This means that the file can be downloaded, regardless of your upload/download ratio.

[T] Transfer File

You will not see this command unless your SysOp has given you access to it. This command will allow you to transfer the file you just viewed from one file area to another. You can also transfer a public file to a private user, or transfer a private file to public with this command. When you select this command, DLG will ask you if the transfer is to be public or private. If the transfer is to be public then DLG will prompt you for the number of the file area. You must have access to an area to be able to transfer a file to it.

[C] Add Comment

This command will enable you to add a comment to the file you have just viewed. Adding a comment to a file is very similar to replying to a message. You will use the same text editor that you use in the message base to write a comment. A comment should be brief and to the point. You could use a comment to point out an error in the description, expand on aspects of the file that you feel the description left out, or express an opinion on the file so that other users can weight the merits of downloading it. Comments are appended to the end of the description text.

[O] Edit Signature

A signature is a personalized text file that DLG will append to the description of each file that you upload. The SysOp may disallow signatures in some or all file areas. You use the same editor to work with your signatures that you use to enter messages.

[.] Tag File

After you read each file description, you have the option of downloading the file you have just viewed, or "tagging" it to download in a batch operation, later. When you tag a file for downloading, it is entered into a list of tagged files that DLG maintains for you. If you want, you can tag several files on one visit, and download them on another visit. Or, if you have downloaded some files from your tagged file list, and the connection becomes interrupted for some reason, the remainder of your tagged list will still be tagged on your next visit.

You can list the contents of your tagged file list by using the "[L] List Tagged" command. You can also clear your tagged file list, or remove a particular file from the list, by using the [Z] Remove Tagged command.

When you have tagged files, the "[D] Download" command works differently. Usually, when you use the download command, DLG will send you the file whose description you have just read. If you have files in your tagged list however, DLG will ask you if you want to download those files instead. If you answer "Yes", then DLG will give you the option of hanging up after the transfer, and will then send you the tagged files. If you answer "No" then DLG will send you the file that you have just viewed.

Reminder: to be able to use the batch file download features of DLG you must be using a file-transfer protocol that supports batch file transfers. We recommend Zmodem, as it is the fastest and most trouble free of the file-transfer protocols.

[L] List Tagged

This command will allow you to see a list of all the files that you have tagged for batch download with the "[.] Tag File" command.

[Z] Remove Tagged

Sometimes when you have tagged files, you want to remove a file from the list, or clear the entire list. When you select the "[Z] Remove Tagged" command, DLG will ask you if you want to clear the entire list. If you answer "No", then DLG will show you your tagged list of files, and prompt you for a filename to remove from the list. DLG will continue to re-prompt you for filenames, until you press RETURN at an empty prompt. This will end the "Remove Tagged" session.

[A] Change Areas

This command has two usages:

1) If you select just the "A" command, DLG will prompt you for the number of a file area to switch to. If you press RETURN at that prompt, DLG will list all the file areas that are available to you. DLG will again prompt you for the number of an area to switch to. If you press RETURN at this prompt, you will remain in the current area.

2) If you select the "A" command followed by a number, DLG will take you directly to that file area if it is available to you. For example, if you type "A1", DLG will take you directly to file area 1.

[S] Select SIG

This command has two usages:

1) If you select just the "S" command, DLG will prompt you for the number of a SIG (Special Interest Group) to switch to. If you press RETURN at that prompt, DLG will list all the SIGS that are available to you. DLG will again prompt you for the number of a SIG to switch to. If you press RETURN at this prompt, you will exit the SIG you are currently in, and remain outside all the SIGS, making all file areas available to you.

2) If you select the "S" command followed by a number, DLG will switch you directly to that SIG, if it is available to you. For example, if you type "S1", DLG will take you directly to SIG number 1.

[N] Next Area

Under your "User Options" from the Main Menu, you can set up a list of areas to "scan" for new files each time you call in. When you first joined this system, DLG placed all the areas that were available to you in your "Global File" list. Additional areas will be added to this list as your access is upgraded. You may add and subtract from that default list at any time by visiting the "User Options" editor available from the Main Menu. This command will take you from the current area to the next area listed in your "Global File" list. DLG will scan through each area in your list and only stop at those that have new files.

TIP: When viewing files in DLG, all you have to do is HIT RETURN. Return will take you to each new file as you read descriptions. As you hit the highest file available in an area, hitting return will do the same thing as using the "N" Next Area command, and will take you to the next area in your "Global File" list that has new files in it. All you have to remember is to "keep hitting return" and DLG will show you all the files in all the areas that you are interested in scanning.

[=] Global New

This command functions identically to the "[F] File List" command, except that it will show files from all file areas that you have in your "Global File" list. You can include and exclude file areas to list with this command by visiting the "User Options" editor from the Main Menu. See the help file for "[F] File List" for detailed information on the options this command will give you.

[P] Private Files

When you enter the File Base, DLG checks to see if you have any private files waiting for you. If you do, DLG will divert you to your private file directory so you can download them right away. Unlike private messages, DLG will not keep track of any "high-file" pointer in your private directory. This is to encourage you to download and delete any private files that you have, as soon as possible. As long as you have a private file in your private file directory, DLG will divert you so that you will be reminded to deal with it.

This command is useful if you wish to visit your private file area to delete files that you have previously downloaded, or if you wish to upload a file privately to another person. All files that you upload when you are in your private file area are automatically private files.

[V] Validate

This command is only available to you if you are a SysOp, or if the SysOp has given you access to this command. When you designate a file area as "validation" area, the files that are uploaded are re-routed to that area. DLG keeps track of where the file "belongs", and gives you access to this command when you view a file that has been re-routed in this manner. If you decide to "validate" a file that you have just viewed, select this command. DLG will prompt you for the number of an area to validate the file to, with the original upload area set as a default.

[^] To Mess

This command will allow you to jump from the file base directly to the message base without having to traverse the Main Menu.

[M] Exit

This command will return you to the Main Menu, where you can explore other DLG sub-systems.

[G] Goodbye

This command will end your current session, and log you off the system.

[H] Help

This command will give you general guidance about the DLG file base, or help on a specific command.

Tutorial - Uploading a File

In this tutorial you are going to upload a file into your new auto-access file area. Since you are going to be "uploading" a file from the same machine the BBS is running on, you will not, of course, actually transfer a file. A local "upload" is simply a way of adding a file and an appropriate description to any of your file areas. If you have many files to add (for example, if you were going to put the contents of a CD ROM on-line, or if you are upgrading to DLG from an existing BBS setup) we suggest that you use the "Turbo" or "Batch" upload options from the File Maintenance section of the SysOp Menu, rather than use this one-at-a-time approach. There will be more on that in a later tutorial.

Make sure you are in the file base of your DLG system. If you followed the previous tutorial on creating file areas, you should already be in the DLG file base.

Choose the Tutorial file area (A99)

Select the Upload Command (U)

The upload will not be private (N)

Enter the name for the file (DLG_UserManual)

Enter the pathname for the file to upload (DLGConfig:Text/BBS-Manual.Txt)

If the name of the file is the same as the actual name of the original file, you may simply enter the path to the file, as long as it ends with either a ":" or a "/" character. Note that the name of the file in your file area does not have to be the same as that of the original file, as in this tutorial example.

Once the file has been copied, DLG will flash the screen, and ask you to provide a description of the file you just uploaded. DLG pauses at this point, by asking you to press RETURN, so that you have a chance to see this notice. This is so that users who are batch uploading several files at a time get a chance to see what file it is that they are describing. Press RETURN, and DLG will put you in the text editor. This is the same text editor that you used in the Message Area tutorial.

Enter the following description:

[R] Read File

Some files that have been uploaded are text files. This command will allow you to view the text contents of a file. Before DLG will read the contents of a file to you, it performs a simple test to determine if, in fact, the file is a text file. If it encounters no non-text characters in the first 20 characters of the file, DLG considers the file to be a text file.

If you have just viewed a file, then the "[R] Read File" command will attempt to show you the text contents of that file. If you have not viewed any file, then DLG will prompt you for then number or a filename of a file to read. You can only read files in the current file area.

[K] Kill File

If you have just viewed a file, then the "[K] Kill File" command will allow you to delete that file from the file area. This command is only active if you have uploaded the file and if the SysOp has granted you this privilege in this file area, or if you have SysOp access in this file area. When you have SysOp access and delete a file that was not uploaded by you, DLG will ask if you want to remove the upload credit for that file.

[I] Inspect Arch

If you have just viewed a file, then the "[I] Inspect Arch" command will attempt to show you the contents of the file, if it is an "archive" file. An archive file is one that has been compressed with an archiving program. Lharc, Zoo, Arc, LZ, and Zip are common archiving programs. You can often tell that a file is an archive by the "extension" on the file name. You will see extensions like ".LZH" or ".ZOO" or ".ARC" or ".ZIP", and so on. DLG will attempt to show you what files are contained within an archive, so that you can determine if it contains a file that you need to download.

[E] Edit File

This command will only be available to you if you either have user level 255 (SysOp), or if the SysOp has given you this privilege in this file area. This command will allow you to "edit" many attributes of a file. This command is identical to some of the file maintenance tools that you will find under the SysOp Menu.

With this command you can edit the following attributes of a file:

[S] From

Allows you to edit the name of the person who uploaded the file

[F] Filename

Allows you to change the name of the file.

[D] Date

Allows you to change the upload date of the file.

[#] Downloads

Allows you to change the download counter on the file.

[K] Size

This command will rescan and update the filesize.

[A] Attributes

This command allows you to adjust a file between one that is a "free" download, and one that requires that a user's upload/download ratio requirements be met.

[B] Edit Body

This allows you to edit the description of the file, including the appended comments.

You will be placed in the now familiar full screen editor, where you may compose a message that will be appended to the end of the file description. Type the following message:

I recommend that all first time users download and read this file. It should clear up any difficulties they might have with using this BBS.

Once you have finished the message, type **CTRL-Z** and then type **"Y"** to save the comment.

Re-display the description for the file (1)

You will see that your comment has been appended to the end of the file description.

```
TL0: JAMES HASTINGS-T B:19200 ON:23:55 TA:1389 T:Aniga
(U,D,F,A,S,N,=,P,T,C,O,,R,K,I,E,^,@,M,G,H,?,RET)
Area: [99] [Tutorial] [1/1-1] => 1
[From ] James Hastings-Trew
[Filename] DLG_USERMANUAL [File 1 OF 1]
[Size ] 83352 [Sat 10 Jul 92 0:03]
[Downls ] 0 [Dwnld Tm] n/a
Downloadable version of DLG's on-line user manual. Download this and print it
out using your favorite word-processor.
----
Comment from user [James Hastings-Trew] on [Sat 10 Jul 92 0:04]
----
I recommend that all first time users download and read this file. It should
clear up any difficulties they might have with using this BBS.
(U,D,F,A,S,N,=,P,T,C,O,,R,K,I,E,^,@,M,G,H,?,RET)
Area: [99] [Tutorial] [1/1-1] => █
```

Any number of users can append comments to the same file. Occasionally a comment will be placed that might be inappropriate, or you might want to take the meaning from the comments and modify the original file description to include the other points of view about the file. See the next tutorial on how to edit a file's description to learn how this is done.

Tutorial - Editing a File

In DLG, there are two ways to edit a file description. One way is to use the File Maintenance module available from the SysOp Menu, and the other way is to do it directly from within the file base itself. In this tutorial, you will learn how to perform this direct type of editing.

This tutorial assumes that you have completed the previous tutorials, and that you are still in your file area 99, the "Tutorial" area, and that you have uploaded a file, and added a comment to that file.

Select file number 1 (1)

Select the Edit File command (E)

This puts you in file edit mode, and your menu will be replaced with one that has the following commands available:

[S] From

This command allows you to change the name of the uploader.

[F] Filename

This command allows you to change the name of the file.

Downloadable version of DLG's on-line user manual. Download this and print it out using your favorite word-processor.

Once you have entered the description, press CTRL-Z to exit the editor. The editor will ask you if you wish to save the message.

Save the file description (Y)

Press RETURN, and DLG will show you the file you just uploaded, and its description.

Note from this tutorial that you can upload files from anywhere on your system - from a floppy disk, your hard drive, or from RAM. DLG makes a copy of the file that you upload in this manner. If you are tight for harddrive space you may wish to delete the original file. Don't delete DLGConfig:Text/BBS-Manual.txt though - it is required to provide the on-line manual option from the Main Menu!

Note also that with DLG you are not restricted to a small file description. A file description can be a whole message in and of itself, and that file areas with their descriptions are not very much different from message areas. The file base does not provide all of the same options for reading and replying to messages that the message base does, but the other functionality is very much the same.

Bear in mind that an on-line user's method of uploading is very similar to this. However, there are a couple of differences:

- An on-line user will have to select a file transfer protocol for moving the file from their machine to yours through the modem connection. A user can select a default transfer protocol when they join your system, or through the User Options from the Main Menu. If they have no default transfer protocol selected, then DLG will prompt them to pick one from a list each time they attempt to upload a file.
- DLG tracks partial uploads (if the file is uploaded with a protocol that allows for a "resume transfer" option) so that if a modem connection is inadvertently interrupted in the middle of a file transfer, the user can continue the file transfer during a later call. DLG will automatically inform users of a partial transfer and prompt them to continue the transfer on their next visit to the file base. If a file transfer is complete, but the user ends the call before they have a chance to enter a description for the file, then DLG will prompt the user to enter the description on their next visit to the File base.
- If you as the SysOp have assigned a "validation" area to a particular file area, then user uploads are re-directed to that alternative area until you have had a chance to validate the files. If the user has a user-level of 255, then this redirection does not occur, because it is assumed that SysOp's have already ascertained the appropriateness of files they are uploading.

Tutorial - Commenting a File's Description

In this tutorial, we will see how users can comment a file description. A comment is an additional message that is appended onto the end of a file description. The comment may have additional information about the nature of the file, it may try to correct information found in the file description, or it may just express an opinion about the suitability of the file for various purposes. Long distance callers in particular appreciate having full information about a file in its description and comments - it may save them from downloading a file that does not really suit their needs.

This tutorial assumes that you are in file area 99, the "Tutorial" file area, and that you have completed the previous tutorial on uploading a file.

Select the file you just uploaded (1)

The file description will display on your screen. Notice that the Add Comment command (C) is now available in your menu.

Choose the Add Comment command (C)

Tutorial - Downloading a File

In this tutorial, we will learn how to download a file from a file area on DLG. As in the tutorial on uploading files, you will not see all of the same questions that an on-line user would see, because you are "downloading" the file on the same system that the BBS is running on. However, the procedures are very similar.

As with most things in DLG, there is more than one way to download a file. We are going to examine each of them in turn. This tutorial assumes that you are in file area 99, and have uploaded a file according to the previous tutorials. Just for practice, we are going to upload two more files, so that we have some material to work with.

Select the Upload command (U)

The file will not be private (N)

Enter the filename (Deckbrowser)

Put your DLG installation disk 1 into your DFD: drive

Enter the pathname (Disk1:Install_Files/Deckbrowser)

Enter the following as the file description:

Deckbrowser 1.5 from Innovatronics. Freely distributable player for CanDo decks
created with CanDo 1.0 and 1.5.

Type CTRL-Z and type Y to save this description.

Now, upload a second file using this same method. Give the file the name "MuchMore", give the pathname as "Disk1:Install_Files/mm", enter the following as the file description, and save the description:

MuchMore text reader by Fridtjof Seibert, version 3.0

Now that we have three files in our "Tutorial" file section, let's look at the various ways of downloading them.

Choose the File List command, Forward, All (FFA)

You will see this listing on your screen:

```
Files in area [99] - [Tutorial]
1  DLG_USERMANUAL  82k  Downloadable version of DLG's on-line user manual...
2  DECBROWSER      121k Deckbrowser 1.5 from Innovatronics. Freely distri...
3  MUCHMORE        19k  MuchMore text reader by Fridtjof Seibert, version...

[U] Upload file      [D] Download          [F] File list        [A] Change areas
[S] Change Sig      [N] New scan          [=] Global list      [P] Private area
[T] Transfer file    [C] Add comment       [O] Edit signature   [,] Tag file
[R] Read file        [K] Kill file         [I] View archive     [E] Edit file
[^] Goto msg area    [M] Send message      [M] Main menu        [G] Goodbye
[H] Help             [?] Display menu      [RET] Next file

Area: [99] [Tutorial] [1/1-3] => █
```

Select the Download command (D)

Enter the name of the file or file-number to download (Muchmore)

Enter the path to download the file to (RAM:)

DLG will "download" the file "MuchMore" to your RAM: drive, by copying the file there. This is the simplest type of downloading available in DLG.

Now, look at the description for the first file (1)

[D] Date

This command allows you to change the date that the file was uploaded on.

[#] Downloads

This command allows you to modify the number of times the file has been downloaded.

[K] Size

This command tells DLG to re-scan and adjust the size of the file to reflect it's actual size. This is useful if you replace a file with a different version by simply replacing the old one in the file base directory with a file of the same name.

[A] Attributes

This command allows you to adjust a file between one that is a "free" download, and one that requires that a user's upload/download ratio requirements be met.

[B] Edit Body

This command allows you to directly edit a file's description message. The file description will be loaded into your editor, where you can make necessary changes.

Select the Attributes Command (A)

You will see this question: **Override ratio [y/N] => Type "Y" for YES.** You have just changed the file from being one that requires a user to have satisfied their upload/download ratio requirements to one that is a "free" download. A file upload/download ratio requires that a user upload a certain amount of file data before they are allowed to download any files. This is a way that some SysOps have of ensuring that their file areas remain active. By making a file free, any user will be able to download this file without needing to worry about their upload/download ratio.

Select the Body command (B)

You will see this question: **Delete old description [y/N] => Type "N" for No, and press RETURN.**

The file description will be loaded into the DLG full screen editor, and you will be able to make the following changes:

If you have typed the descriptions and comments exactly as presented in the previous tutorials, then enter the following keystrokes:

Press the DOWN cursor key three times to move the cursor down to the first line of the comment divider line.

Type CTRL-Y four times to delete four lines from the message.

You should end up with a file description that reads as follows:

Downloadable version of DLG's on-Line manual. Download this and print it out using your favorite word processor.

I recommend that all first time users download and read this file. It should clear up any difficulties that they might have using this BBS.

Once you are satisfied with the file description, **type CTRL-Z and type "Y" to save the new file description.**

Press RETURN to exit from the File Edit mode.

Select file number 1 (1)

DLG will show the newly edited file description. Note that the file is now marked as a "free" download.

If you have your file areas set up to redirect user uploads to a validation file area, the original area that the file was uploaded to will be "remembered" by DLG, so that when you validate the file, DLG can automatically place the file in the area that the user uploaded it to. You also have the choice of validating the file to a different area than that to which it was originally uploaded.

Occasionally it will come to your attention that a file has been inadvertently or mistakenly uploaded to the wrong file area. You can use the [T] Transfer File command from the File Menu to move the file to a more appropriate area. DLG will prompt you for the number of the file area to move the file to. You can optionally allow users the ability to transfer their own files as a default attribute when you create the file areas so that users can correct their own mistakes.

It is possible that you are upgrading to DLG from a previous BBS package. If that is the case, you probably already have established file areas full of uploaded files. If this is your situation you will find that the file maintenance features built into the file base will be inadequate to the task of moving large numbers of files into your newly created file directories. Alternatively, you may wish to put the contents of a large volume, such as a removable hard-drive or CD ROM on line, and you do not wish to actually copy the files from one volume to another just to provide the files for downloading. DLG provides you with all of the tools to work with these situations, and more, in the File Maintenance section of the SysOp Menu.

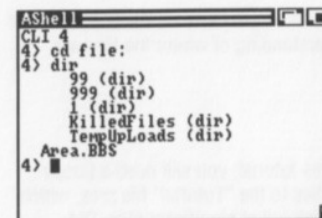
Before we can discuss the various options for batch uploading and maintenance of files, we need to discuss some of the technical aspects of how DLG stores files for its file areas.

When you create a file area in DLG in the File Area Define / Edit section of the SysOp Menu, DLG will create a directory in the FILE: volume for that area. The directory is named for the number you give to the file area.

Type the following in your CLI or Shell:

```
dir file:
```

You will see the following:



```
AShell
CLI 4
4> dir file:
4> dir
  99 (dir)
 999 (dir)
   1 (dir)
KilledFiles (dir)
TempUploads (dir)
Area.BBS
4>
```

The directory named "1" is the "General" file area that the DLG installation utility created for you, the directory named "999" is the SysOp validation area that you created in the tutorial on creating file areas, and the directory named "99" is the "Tutorial" file area that we have been working with in the last few tutorials. The directory called "TempUploads" is the place that all files are uploaded to. When a user uploads a file, a directory under that user's name is created inside the "TempUploads" directory, and the files are uploaded there. The directory called "KilledFiles" is the place that files are put when you delete them from your file base. It is important to know this, as **the hard drive space is not recovered from an uploaded file when you delete it from your file areas**. You will need to periodically go into this "KilledFiles" directory and physically delete the files from your hard drive to reclaim the drive space.

Type the following in your CLI:

```
cd file:99
dir
```

Select the Download command (D)

Enter the path to download the file to (RAM:)

With this method of downloading, DLG allows you to download each file as you read the file descriptions. As you enter the file base and read each new file description, you have the option of immediately downloading that file, or pressing RETURN to view the next file description. This is a very simple and direct way to download files, as you do not have to remember the file name or file number in order to download the file.

Now, let's examine the third way of downloading files with DLG.

Look at the description for the first file (1)

Tag the file for downloading (.)

Press RETURN to see the next file description

Tag the file for downloading (.)

Press RETURN to see the next file description

Tag the file for downloading (.)

At this point, you have three files marked for download. You can mark as many files as you like, from many different file areas. DLG will remember the entire list until you either remove them from the list, delete the list, or successfully download each of the files on the list. Let's look at the list of files that we have marked:

Select the List Tagged command (L)

DLG will list the three files that you have marked for batch download. Let's say that we wish to change our minds, and not download one of the files:

Select the Remove Tagged command (Z)

You will see this question: Would you like to delete the entire list [Y/n] => (N)

DLG will list the files you have tagged, and you will see this question: Enter filename of file to un-tag [Return to abort] => (Deckbrowser)

DLG will relist the remaining tagged files for you. Press RETURN to exit from "Removed Tagged" mode.

Select the Download command (D)

You will see this question: You have pending tagged files. Do you wish to download them instead [Y/n] => (Y)

Enter the path to download the files to (RAM:)

DLG will download all of the files to your RAM: drive.

You can tag files during many different visits to the DLG file base, and from any available DLG file area. DLG will remember all of your tagged files for you so that you will be able to download them on subsequent visits. DLG will even track files that have not been successfully downloaded, keeping them in the list until you are able to fully download the file. This is a powerful convenience feature for your users.

File Area Maintenance Overview

In the previous tutorials we have seen how to upload and download files, and even perform limited file maintenance on files in your file areas. This section will cover some aspects of DLG file area maintenance not covered in the tutorials.

This is the name that will appear in the file header. This is to say who the file is from. Usually you will put your name, or SysOp.

Mark as free [y/N] =>

A free file is one that is free to download regardless of a user's file upload/download ratio. This attribute is given to all of the files that you turbo upload.

Move files to file area [Y/n] =>

If you answer YES to this question, the files will be copied from their source directory to the file area directory itself. **The files will be deleted from the source directory as they are copied.** If you answer NO to this question, the files will remain in the source directory, and not be copied to the file area directory.

Note that if you answer NO to this question, the files must actually be in the path for the file area in question. For example, if you specified DH1: as the alternate path for file area 99, then the files that you are going to turbo upload with NO MOVE should already be in the root directory of DH1: Most commonly this would be used in conjunction with a "global path."

**Enter default description
or press [RETURN] to take descriptions from filenotes**

This allows you to either provide a default description for all the files that you want to upload, or to take the file description from each file's filenotes. A filenote is a method of attaching a comment to a file. Some BBS programs store the file description in the filenote of each file, so this is a handy way of upgrading a file area from such a package to DLG.

Path to source files =>

This allows you to indicate where the source files are. If you told DLG to move the files to the file area, then the source files will be deleted as they are copied over.

[B] Batch Upload

This is nearly identical in operation to the Turbo Upload feature, except that you will be prompted to provide an individual file description for each file found in the source directory. All other operations are identical to the Turbo upload function.

[L] Freshen

This command will "freshen" a file area, in case it's files become out of sync with the actual file area contents, or if the file area's control files become damaged.

[A] Select Area

This command will allow you to select a file area to work on. Normally, when you enter the Transfer Maintenance module you will be working with the last file area that you were in when you were in the file base. You can use this command to list the available file areas, and then pick one to work with. Most commands in the Transfer Maintenance module work on the currently selected file area.

[F] List Files

This command will list files in the currently selected file area.

[E] Edit File

This command will allow you to edit files in the currently selected file area. This file editing ability is identical to that found within the file base itself. You can edit the filename, uploader name, upload date, file size, file attributes, and the file description.

You will see the following:

```
AShell
CLI 4
4) cd file:
4) dir
  99 (dir)
  999 (dir)
  1 (dir)
  KilledFiles (dir)
  TempUploads (dir)
Area.BBS
4) cd file:99
4) dir
  1.fd
  2.fd
  3.fd
  DLG_USERMANUAL
  MUCHMORE
  User.File
  DECKBROWSER
  File.dat
  Pointers.file
```

The files named with an ".fd" extension are the file descriptions for each of the files in the file area. You will also see the actual files in that same directory. The file "File.dat" contains an index of the files with their "quick list" descriptions. The file "user.file" contains the names of the people who are active in this file area, and a pointer indicating which was the highest file number they have seen in this area. The file "Pointers.file" contains the lowest and highest file numbers for this area.

When you created this file area, you were asked if you wished to name an alternate path for this file area. In the tutorial, you were instructed to use the default path for this file area. This means that the uploaded files are contained in the same directory as all of the DLG control files for this area. If you had entered a different path name, then all of these control files would still be in this directory, but the actual uploaded files themselves would be on the alternate volume. That is to say, the ".fd", "file.dat", "user.file", and "pointers.file" files would still appear in the directory on the FILE: volume, but the files "DLG_USERMANUAL", "DECKBROWSER", and "MUCHMORE" would appear on the alternate path for the file area.

It is important to understand these different ways of storing files in a DLG file area, because some of the options in the File Maintenance module depend on this understanding of where the files are actually going to be physically located.

Tutorial - Batch Uploading Files

In this tutorial, we are going to batch upload several files. For this tutorial, you will need a floppy diskette that has several files on it. We will be uploading these files to the "Tutorial" file area, which we will delete in a later tutorial, so don't worry about finding a disk full of significant files. Old wordprocessor files, pictures, utilities, and so, will do nicely.

To begin this tutorial, go to the SysOp Menu. If you have followed along with the previous tutorials, you are in the file base at this moment.

Go to the Main Menu, and then to the SysOp Menu (MS)

Select the Transfer Area Maintenance command (T)

You will be presented with a new menu of commands. They are listed here with a brief explanation:

[T] Turbo Upload

The Turbo Upload command gives you the option of adding many files to a DLG file area at once with a minimum of operator intervention. You specify a path to the source files, and all files in that path will be turbo uploaded to the current file area. You will be presented with a number of prompts to guide you through the process:

Name of uploader =>

At this point, having worked through these tutorials, you should be getting familiar with the DLG environment. Therefore, from now on, you will be instructed to go to a particular area of the DLG system, but not given specific instructions on how to get there. For example, if we say "go to the SysOp Menu", you should know by now that the SysOp Menu is available from the Main Menu of your DLG system. Remember that the "M" command usually takes you back one menu level. The exception to this rule is the "M" command from the Main Menu - that takes you to the Message base.

In this tutorial we are going to create another temporary tutorial file area.

Go to the SysOp Menu, and select the "File Area Define / Edit" command. Follow through the prompts, and create a **special access area**. Give the area number **"1000"**. Name it **"CD ROM Tutorial Area"**.

You can simply hit return at most of the following prompts to select the defaults. For upload access, enter **255** for both values, as we do not want to allow any upload access to this file area. By restricting this to only you, the SysOp, we can prevent users from trying to upload to this area.

When you get to the prompt that asks for an alternate path for the area, type "DF0:".

We are going to pretend that the write-protected disk in drive DF0: is a CD ROM, which is really just a large non-writeable disk volume as far as the Amiga is concerned.

Once you have successfully created this new file area, **exit the File Area Editor and enter the Transfer Maintenance module. You can do this quickly by typing "MT" and hitting RETURN.**

Type "A1000" to select your new file area as the current area to work with.

Type "T" to select the turbo upload mode.

Enter "SysOp" as the name of the uploader.

Press RETURN to select the default for the next prompt.

You will see this question: **Move files to file area [Y/n] =>.** Type **"N" for No.** We want to leave the files on the "CD ROM", not copy them to our hard drive. Since we set the alternate path for this directory to the "CD ROM", DLG will be able to find the files properly.

Enter a default file description of "CD ROM file."

Enter "DF0:" as the source path.

DLG will turbo upload all of the files in the DF0: directory, assigning them the default file description you entered.

CD ROM guidelines

Not many CD ROMS have all their files in the root directory. Lets look at a typical example, and see what you have to do in order to get this volume to work on your system:

Let's say that you have a CD ROM that has three main directories: Graphic, Sound, Anim. All of the files in each category are loose in their main directories - files in subdirectories will not be uploaded, and must be handled separately. You will have to make a short assignment for each of these directories, because the alternate path you give DLG for each file area can only consist of 6 characters. You will need to add these assignments to your S:DLG.Startup file so that they are active each time your start up your system. For example, you would create the following assignments (these examples assume that the CD ROM has the device name of CDO: - substitute the real device name of your particular CD ROM device).

```
assign GRA: CDO:Graphic
assign SOU: CDO:Sound
assign ANI: CDO:Anim
```

[G] Global Edit

This command will allow you to adjust the upload/download or free attribute on every file in the current file area.

[M] Main Menu

This command will return you to the SysOp Menu, and end the Transfer Maintenance session

[?] Help

This command will provide simple on-line help for each of these commands (listed, for the most part, above).

Now, let's see the turbo upload feature in action. Select a floppy diskette which has a number of small files on it. Make sure that you have the write protect tab set on the disk so that the files cannot be deleted. Insert the disk into drive DF0:

Select the Tutorial File area to work on (A99)

Select the Turbo Upload command (T)

You will see this question: Name of uploader => (SysOp)

You will see this question: Mark as free [Y/N] => (N)

You will see this question: Move files to file area [Y/n] =>. Press RETURN to select the default of YES. Note that default responses in DLG are always indicated at a prompt by being in upper case.

You will see this prompt:

Enter default description

or press [RETURN] to take descriptions from filenotes

Type "A file for your downloading pleasure" and press RETURN. You are allowed to enter a simple description of up to 254 characters at this prompt.

You will see this prompt: Path to source files =>. Type "DF0:" and press RETURN.

DLG will then proceed to upload all of the files that it finds on the root directory on the disk in DF0:. It will copy the files from DF0: and place them in our tutorial file area, giving them the standard file description and other information that you provided.

Lets go to the area and see what happened. **Type "MMFA99" to exit the Transfer Maintenance module, exit the SysOp Menu, go to the File base, and enter file area 99.**

Press RETURN, and you will see the first of the files that you just turbo uploaded. **Continue pressing RETURN**, and you will see each turbo uploaded file, in turn until you come to the end of the new files.

Tutorial - Uploading files from a CD ROM

Now, let's see how to create a file area for a CD ROM situation, and turbo upload the files from that device. Since most SysOps will not have a CD ROM drive, we are going to demonstrate how to do this by using the write-protected floppy that we used in the last tutorial.

The file organization on a CD ROM can vary widely, depending upon the contents of the disk. Files of a similar nature are usually grouped together in subdirectories on the disk. This is convenient, because it makes it easy to create several file areas for the CD ROM, with each area having a particular focus. If the CD ROM you wish to use is cut up into too many little directories, or if the files on the CD ROM are not in an archive format for easy downloading, this will be less attractive to work with.

Once you add all of these global paths, you can create your file areas however you like, and turbo or batch upload the files from several different directories on the CD ROM.

For example, lets say you just create one CD ROM Graphics file area. You would set the area so that default user access would be download only. You would not have to set the alternate path - you could leave it at the default for the area, since no download files are going to be actually placed in that directory anyway. When you Turbo or Batch upload the files, you would work with each directory on the CD ROM in turn, but tell DLG not to move the files to the file area. When a user goes to download a file from this area, DLG will first search the directory in FILE:, but will not be able to locate the download file. It will then search each of the global paths you created for the CD ROM in turn until the file is located for download.

This alternative method of working with CD ROM devices can also be used to work with other large volumes, where you do not want to create additional assigns on your system.

The concept of global paths has other implications. For example, when a file that exists on a global path is transferred from one file area to another, it will be left in place, rather than being moved to the destination directory. Because the file is in a global path, DLG will be able to find it no matter what file area its description appears in.

If you decide to move files from an existing FILE:<areanumber> directory, or alternate path, to a global path by copying or moving the files with the AmigaDOS "copy" command, or with a directory utility, be sure to use the CLONE option. This *must* be done so that the file comments are maintained.

CD ROM Side Effect

DLG maintains a link between a file and it's description by using the file's AmigaDOS filenote. In the case of CD ROM files, DLG cannot modify the filecomment so that it indicates which description is associated with it. This only has a negative effect on batch downloads. Since DLG only "knows" the name of the file to download, and not the description number, DLG will be unable to locate the description for CD ROM files. The file can still be downloaded, but the download counter will not be updated to indicate the correct number of accesses for the file. If the file is directly downloaded from the file area with the "D" command, DLG will update the file header to indicate the download. If a user downloads a file by it's number, the download counter will also be updated, but not if the user downloads the file by name.

Tutorial - Modifying or Deleting a File Area

Now that you have seen most of the SysOp functions for creating, modifying, and maintaining DLG file areas, we will go through the steps required to delete files and entire file areas. First, let's delete the files from our tutorial file area.

Go to the File base, and select area 99 as your current file area.

Type "1" and press RETURN to see the listing for file #1.

Type "K" for Kill file and press RETURN.

You will see this prompt: Kill file [DLG_USERMANUAL] [y/N] =>. Type "Y" for Yes, and press RETURN.

You will see this prompt: Would you like to remove the credit for this upload? [y/N] =>. This is asking you if you would like to remove the upload ratio credit from the user who uploaded this file. Type "N" for No and press RETURN.

Now, go to your CLI or Shell, and type the following:

```
cd File:Killedfiles
dir
```


When creating file areas for these CD ROM directories, simply give the appropriate assignment as the alternate path for each area. Pay particular attention to the fact that you should not allow upload access to a CD ROM area.

Once you have created the appropriate file areas for the various directories of the CD ROM, you can either Turbo Upload the files, or Batch Upload the files if you feel the need to add individual file descriptions for each file on the device. Be sure to tell DLG to not move the files to the file area. If the files on the CD ROM have descriptive filenotes, we suggest you take advantage of that fact, and use them as default file descriptions from the Turbo Upload mode.

Bear in mind that with this arrangement that the particular CD ROM in question will need to be in the CD ROM drive when you start up your system, so that the assignments can be made successfully.

If your CD ROM disk is fragmented into several small directories that have few files in them, then this method would be unworkable. You would not want to create a file area for each small subdirectory on the CD ROM. There is an alternative method that does not require assigns to be made.

If you go into the SysOp Menu, and select the "General Configuration" module, you will find that there is a command called "Add Paths". These paths are "global" file paths that all DLG file areas share. If a file is not found in the path for a file area, then DLG will search all of these "global" paths looking for the file. This can be a great help in creating file areas for a CD ROM that has many small subdirectories. Let's look at an example:

Let's say that your CD ROM has the following directory structure:

```
CD0:
Graphics
  Nature
  Ray_Tracing
  Humor
  Art
  Animals
  Abstract
  Science_Fiction
  Clip_Art
  PD_Painting_programs
  Slideshow_utilities
Animations
Sounds
  Music_Modules
  Utilities
  Sound_samples
    mechanical
    animal
    sound_effects
    miscellaneous
```

etc. Using the General Configuration module from the SysOp Menu, you would add each full directory path on your CD ROM to the global paths list. For example, you would add these paths:

```
CD0:Graphics/nature
CD0:Graphics/Ray_Tracing
CD0:Graphics/Humour
CD0:Graphics/Art
CD0:Graphics/Animals
...
CD0:Sounds/Sound_samples/sound_effects
CD0:Sounds/Sound_samples/miscellaneous
```

You will see that the file DLG-USERSMANUAL was not actually deleted from your hard drive. It is actually moved to the directory File:Killedfiles. Think of this directory as having a similar function to the Workbench Trashcan directory. Files are thrown into this directory, but may be retrieved at a later time, until you actually delete the files from this directory yourself. This is important to remember. If you are cleaning up old files from your file areas, you will not actually re-gain the hard drive space until you manually delete the files from File:KilledFiles. This is also true of files that users receive in their private file directories. When they kill the files after downloading them, they are moved to File:KilledFiles. You can use this fact to monitor private file activity on your system, to ensure that users are not using private files to transfer copyrighted materials to each other. If you type the following in your CLI:

```
cd File:KilledFiles
List
```

you will see a listing of all of the files that are there, with the name of the recipient of each file listed in the filenote. If you see copyrighted material being exchanged, you can take disciplinary action based on this information.

Now, let's get rid of the tutorial file areas that we created in this chapter.

Go to the SysOp Menu and select the "File Area Define / Edit" command.

Type "D" for "Delete Area" and press RETURN.

You will see this question: Delete which file area =>. Type "99" and press RETURN.

You will see this question: Delete file area 99: Are you sure [y/N] =>. Type "Y" for Yes, and press RETURN.

DLG will delete the file area and all of the files associated with that directory.

Follow these same steps to delete file area 1000.

Conclusion and Tips

You should now know how to create DLG file areas, and have a working knowledge of what some of the characteristics of those areas can be set to. At this point, take some time and plan out what kind of file areas you will want to have on your system. Here are some tips to consider when creating your files areas. Many of these tips are identical to those offered in the last chapter on creating message areas:

1. The majority of your file areas should be auto-access. Control access to those areas by managing user-levels. This is a much simpler method than trying to manage special-access file areas. Reserve special-access areas for those whose membership does not change very often.
2. Take some time to think about the structure of your file base when you are starting to assign numbers to your file areas. Try to group related areas together by number. Allow yourself room for expansion. For example, you could put one group of areas together starting with 10, another group starting with 20, and so on. If you really want to give yourself room, group the areas by 100s. This makes it easier for you, and your users, to remember where particular areas are likely to be. For instance, all of your Amiga areas could be in the range of 100 to 199. Your Macintosh areas could be in the range of 200 to 299, and so on.
3. If you wish to have file areas associated with message areas of a similar interest on your system, take time to construct the structure of your areas so that similar message and files areas have the same numbers. This will make it easy for users to associate the numbers with the topic of interest.

4. As mentioned above, a user's private file is sent to the File:Killedfiles directory when the user deletes the file after downloading it. You can use this fact to monitor the private file activity on your system. By LISTing the files in File:Killedfiles you can see who the recipient of each file is, and use this information to deal with users who are utilizing your system to exchange copy-righted materials.

Example Section

Q I want to make sure that users upload files, and do not simply download everything on my system without giving anything back. What steps do I take?

A You can establish upload/download ratios for users when you validate them on your system. DLG operates on a byte count upload/download ratio system. This means that if a user's ratio is set to 1/10, they will have to upload 1 byte of information for every 10 bytes they wish to download. For example, a user uploading a 100K file would then be able to download 1 Meg of files before DLG would preventing them from downloading further files. A value of 0 means that no upload/download ratio will be imposed.

You should provide some basic files as "free" downloads on your system. These "free" downloads are available to all users, regardless of the current state of their upload/download ratio. "Free" files also do not count against a user's download byte count. Basic files should include file archiving utilities, the DLG user manual, basic information about your board, etc.

Q I want to set my DLG system up as a pay system. When new users apply, I want them only to have access to a few areas. I want the users to have access to different file and message areas based on a scale of fees. How do I do this?

A Create your message and file areas as auto-access areas, with a scale of access-levels that matches your planned scale of fees. Assign user-levels to users based on the fees that they pay, and they will automatically gain access to the features that they have paid for. If you decide later to demote users because of delinquent fees, then users will automatically lose access to areas as you lower their user-levels. This is much better than trying to create special-access areas and trying to add and delete users individually.

Q I want to be able to designate different users on my system as "area SysOps," to help keep track of areas that I might not have time to look after properly. But, I don't want them to access other areas, or the SysOp menu, so I can't do this by adjusting their user-levels. What do I do?

A Set up your file areas as you normally would, as auto-access areas. If the areas in question have existed for some time, then all you need to do is edit the users with the "Revise File Area Access" editor. For each "area SysOp," you will want to add access in the area they are to look after. You will add these as overrides to the existing defaults in the areas. If you have just created the areas, there will be no users present in the areas to edit. You will have to add your intended "area SysOp" to the area you just created, and then you will be able to edit the access.

Q I need a file area for myself and all of my other "area SysOps" to share files, and provide a central place for newly uploaded files that need validation. The problem is, they all have various user-levels, and I don't want other users to be able to access the files in that area. How do I set this up?

A This is the perfect situation for a special-access area. The membership is not going to be fluctuating, and new members are not going to be added very often. The area has a special function, and the membership is selected across the spectrum of your user base. You would create the area as a special-access area, and then individually add each user that you wish to include in that area with the "Revise File Area Access" editor.

Reference Section

This section contains a detailed discussion of DLG file areas. This information is provided as reference material, and contains some duplication of information already covered in the tutorials in this chapter. If you are just starting to set up your DLG system, please skip ahead to the next chapter on working with User Accounts.

DLG File Areas

DLG's File Areas are stored in the assigned volume FILE:. If you list a directory of FILE: you will see a number of other directories listed. When a DLG file area is created, a directory for that area is created in the assigned volume FILE:. The name of the directory is the number of that file area. There are two other directories that are present in the FILE: volume:

- TempUploads - when a user uploads a file, a temporary directory is created in File:TempUploads with this name used as the directory name. All of that user's uploads are held in this directory until the user completes the upload and provides a full description of the files that they have uploaded. Each time the user visits the file base, and has unfinished uploads, or uploads that they has not yet provided a description for, they will be reminded of that fact and prompted to either finish the upload, delete the file, or provide the necessary file description.
- KilledFiles - when someone kills a file in a public file area, the file is not actually deleted - it ends up here in the KilledFiles directory. You will have to remember to delete these files from time to time, or you may wish to set up a TPTCron event to do this on a periodic basis. Files that were uploaded privately to users also end up in this directory. You can ascertain who the file was uploaded to by using the AmigaDOS LIST command in the File:KilledFiles directory.

In addition to these directories you will also see a file called Area.BBS. This file contains the descriptions and default user settings for each of the file areas that you have created on your board.

A file area directory will contain a number of control files. These are files used by the DLG system to help manage the download files present in the file directories. The most numerous of these files will be the file descriptions. These have the name "n.FD" where "n" is the file number. You will also see the uploaded files themselves, unless you have used the optional assignment to place a file area's files on a different volume, or if you have placed the files for the area on one of the global paths that you can define in the General Configuration Editor. Each file has a FILENOTE to indicate the number of the file description associated with it. This allows DLG to easily determine which file description to move or delete when the SysOp elects to transfer or kill a particular file by name. The other files that you will see are:

- Pointers.FILE : This file contains the high and low file pointers for that area, and will be reset by the RENUMBER program when it is run on the file areas as well as the DLG file sub-system to determine how many files are in the current area.
- User.FILE : This file contains a list of the users who have access to this area, and what kind of access they have. It also contains the high file pointers for each user for this file area.
- File.DAT : This file is an index of all the files in the file area. It contains the "quick" list of files, and is used by DLG to provide all of the various file listing and search operations available through the "File List" command in the file base. If this file becomes corrupted for any particular reason, you can use the "Freshen" command in the Transfer Maintenance Editor to rebuild this index.

In many ways, DLG treats file areas much like message areas. The creation of file areas is accomplished in a similar fashion - you provide the number of the file area you wish to create, and DLG prompts you for various parameters. You create/edit file areas with the "Edit/Define File Areas" command from the SysOp Sub-System. Here is a list of the file area parameters:

Area Number: This is the number by which the file area will be shown in the file area list. This number is also the name of the directory in FILE: where the control files for the file area exist, and may also be the directory where the download files themselves reside.

Area Name: This is the name by which the file area will be shown in the file area list. The name should be descriptive of the kinds of files that will be found in that file area.

Auto Access Area: This attribute controls if users can gain access automatically to the area, based on user level, or if they need to be manually added to the area by the SysOp. If a file area is not auto-access then it is called a special-access area. The access to an area is controlled by two threshold values - a lower user level, and an upper user level. Only those users whose levels fall between the lower and upper thresholds will have automatic access to the area. A user with level 255 can enter any file area, regardless of the threshold levels.

There are a number of file area attributes which determine what kind of activities will be available to users in a file area that you create. Like the entry access levels, these consist of lower and upper threshold values, keyed by user-level:

Upload privileges: users whose levels fall between the lower and upper threshold levels will be able to upload files to this area

Download privileges: users whose levels fall between the lower and upper threshold levels will be able to download files from this area.

Transfer privileges: users whose levels fall between the lower and upper threshold levels will be able to transfer files that they have uploaded from this file area to any other file area that they have access to.

Kill privileges: users whose levels fall between the lower and upper threshold levels will be able to kill files that they have uploaded to this area.

SysOp privileges. users whose levels fall between the lower and upper threshold levels will be able to transfer, edit, validate or kill files in this area, even if they are not the uploader.

There remain a few other attributes of the file area:

SysOp Validation: If this is set, then files that are uploaded to this file area will be shunted off to a different file area, until the SysOp or a Co-SysOp "validates" the file. At the time of file validation, the file will be returned to the file area it was originally uploaded to. DLG will ask you if the SysOp needs to validate new uploads. Type "Y" and press RETURN.

Validation Area: If you tell DLG to define the area as one that requires SysOp validation of newly uploaded files, then it will prompt you DLG will prompt you for the number of a different file area to re-direct the new uploads to. If you answer the "validate" prompt "No," then this question will be skipped. Many different file areas can share the same validation area - DLG keeps track of where files have originally been uploaded to.

File Requestable: A file request is a FidoNet procedure by which a system can call up your system, and pass to it a list of files to be sent. DLG, in concert with appropriate "front-end" software, can fulfill these file requests. However, if you do not want, or need, this capability in a particular file area, you can turn it off.

Alternate Path: Normally, the files in a particular file area are stored in the same directory that the file area itself is contained in. However, if you have extra hard drive volumes or partitions that have lots of room, you can tell DLG to store the files on the alternate volume or partition. Bear in mind that the alternate path can only consist of six characters, including any ending ":" or "/" character. If the path that you wish to use here is longer than that, you will have to create an assign for that path. Note that you do not have to specify the "/" character if the path is indicating a directory on the alternate volume - the software will automatically append it for you.

Other Options That Affect DLG File Areas

When files are uploaded to DLG file areas by users, they are stored in the same directory as the file area control files, or they are stored in the alternate path that you give to the file area. You may also have files available for downloading that are not contained in either of these locations, by defining global file paths. This is done through the General Configuration Editor from the SysOp Menu. If you Turbo or Batch upload files to a DLG file area from a global file path, or if you manually move the download files from the file directory or alternate file path to one of the defined global file paths, the DLG software will still be able to find that file when a user attempts to download it.

For example, let's say that many users have uploaded files to your "Music Module" file area, and the hard drive where FILE: is located is getting full. You have another hard drive that has more room. You can specify that other hard drive as a global file path, and move the files from the file area directory to that other hard drive. DLG will still be able to find the files on that global path. **If you do move files to a different HD, make sure you use the "clone" option to preserve the file notes and attributes.**

You can use either the alternate path or the global path method to add large volumes, such as CD ROM disks, to your DLG setup.

DLG File Area Text and Batch Files

There are a few text and batch files that DLG will optionally use, if they exist on your system. Here is a list of the available text and batch file options. To use a particular option, you need only create the given text or batch file with a text editor, and place the file in the appropriate directory. DLG will automatically use the file if it finds it. You may use any of the system %switches (see the chapter on DLG Text files for a listing of the available %switches) to imbed user or DLG specific information in the text files:

FILE:<areanumber>/EnterArea.txt

This functions identically to the file of the same name used in DLG message areas. This text file, if placed in the FILE: directory of a particular file area, will be displayed by DLG when a user enters that file area. See the reference section of the previous chapter for more information.

FILE:<areanumber>/Uploadfile.txt

DLGConfig:Text/Uploadfile.txt

These text files will be displayed to the user prior to an upload session. If a FILE:<areanumber>/Uploadfile.txt file is found, it will override the DLGConfig:Text/Uploadfile.txt file.

DLGConfig:Text/RefuseDownload.txt

This text file will be displayed when a user attempts to download a file that is in conflict with their file ratio. You can use this file to explain to the user that they need to upload some files to be able to continue downloading.

DLGConfig:Batch/ReceivedFile.batch

This batch file will be executed once for each file that exists in the upload, with the following parameters: <path/filename> and <filename>.

You can use this batch file for whatever you want - to test the integrity of archived files, or to check for duplicate files, etc.

DLGConfig:Batch/Upload1.batch

This batch file is executed just before a user enters a file description for an uploaded file. It is passed two parameters: <username> and <path/filename>.

DLGConfig:Batch/Upload2.batch

This batch file is executed just after a user enters a file description for an uploaded file. It is passed two parameters: <username> and <path/filename>.

File Area Maintenance Editor

DLG file areas can be maintained with the File Area Maintenance Editor from the SysOp Menu. Here is a command by command description of the functions in the File Area Maintenance Editor. This is a repeat of the same information found earlier in this chapter, and is included in this reference section for the sake of completeness:

[T] Turbo Upload

The Turbo Upload command gives you the option of adding many files to a DLG file area at once with a minimum of operator intervention. You specify a path to the source files, and all files in that path will be turbo uploaded to the current file area. You will be presented with a number of prompts to guide you through the process:

Name of uploader =>

This is the name that will appear in the file header. This is to say who the file is from. Usually, you will put your name, or SysOp.

Mark as free [y/N] =>

A free file is one that is free to download regardless of a user's file upload/download ratio. This attribute is given to all of the files that you turbo upload.

Move files to file area [Y/n] =>

If you answer YES to this question, the files will be copied from their source directory to the file area directory itself. The files will be deleted from the source directory as they are copied. If you answer NO to this question, the files will remain in the source directory, and not be copied to the file area directory.

Note that if you answer NO to this question, the files must actually be in the path for the file area in question. That is, if the path for the file area is at the default, then the files must already be in the file area's directory. If you have specified an alternate path for the file area, then the files must already be on the alternate path. For example, if you specified DH1: as the alternate path for file area 999, then the files that you are going to turbo upload with NO MOVE should already be in the root directory of DH1:

Enter default description or press [RETURN] to take descriptions from filenotes

This allows you to either provide a default description for all the files that you want to upload, or to take the file description from each file's filenotes. Filenotes are an AmigaDOS way of attaching a comment to a file. Some BBS programs store the file description in the filenote of each file, so this is a handy way of upgrading a file area from such a package to DLG.

Path to source files =>

This allows you to indicate where the source files are. If you told DLG to move the files to the file area, then **the source files will be deleted as they are copied over.**

[B] Batch Upload

This is nearly identical in operation to the Turbo Upload feature, except that you will be prompted to provide an individual file description for each file found in the source directory. All other operations are identical to the Turbo upload function.

[L] Freshen

This command will "freshen" a file area, in case its files become out of sync with the actual file area contents, or if the file area's control files become damaged.

[A] Select Area

This command will allow you to select a file area to work on. Normally, when you enter the Transfer Maintenance module you will be working with the last file area that you were in when you were in the file base. You can use this command to list the available file areas and then pick one to work with. Most commands in the Transfer Maintenance module work on the currently selected file area.

[F] List Files

This command will list files in the currently selected file area.

[E] Edit File

This command will allow you to edit files in the currently selected file area. This file editing ability is identical to that found within the file base itself. You can edit the filename, uploader name, upload date, file size, file attributes, and the file description.

[G] Global Edit

This command will allow you to adjust the upload/download or free attribute on every file in the current file area.

[M] Main Menu

This command will return you to the SysOp Menu, and end the Transfer Maintenance session.

[?] Help

This command will provide simple on-line help for each of these commands (listed, for the most part, above).

User Access to File Areas

When you create a file area, DLG creates a directory for that area in the FILE: directory. Each file area has its own set of rules — there are ranges that you can set that define the behavior of that area for each user according to user level. Each file area has a list of members, and each area keeps track of what overrides each member has in that area. Each area also keeps track of the high file pointer for each user, so that users can easily find new files in that area.

This creates a flexible environment for SysOp control. You can set up default access values for users in areas, and you will only have to modify users' individual user-levels to make changes to their access. This entails some responsibilities on your part — make sure that you note which areas require what user level, and what attributes require what attribute level, so that when you create user editing templates, you can easily find and control what you want to do with each user.

If you need to, you can individually adjust the access given to individual users with the "Revise File Area Access" editor in the SysOp menu.

Here is a list of things that you can do in this editor:

[L] List Area

After you select this command, DLG will prompt you for the name of a user. Once you provide this, DLG will list all the areas that the user has access to.

[U] List Users

After you select this command, DLG will prompt you for the number of a file area. Once you provide this, DLG will list all the users who are members of that area, and will show any special access flags you may have set for them.

[A] Add User

This command will allow you to add users to a special-access file area. DLG will prompt you first for the number of the area, and then for the name of the user to add to that area. The prompt for the user's name will keep coming back until you hit RETURN on an empty entry. This allows you to enter several names with this single command.

[D] Delete User

This command will allow you to delete a user from a special-access file area. DLG will prompt you first for the name of the user to delete, and then will prompt you for the number of the area to delete them from.

[C] Area Copy

This command will copy the "user base" from one file area to another. Whichever users have access in one area will end up with the same access in another, along with their special-access flags. This is a convenient way of creating several special-access areas that contain similar memberships.

[E] Edit User

This command will allow you to directly edit the access of one user in one file area. You can set special access flags that will override the user-level ranges for that area. DLG will prompt you for the name of a user. Then DLG will prompt you for the file area to edit that user's access in. DLG will then display the following prompts:

```
[+] Add [-] Remove [RETURN] No change
      Upload access: =>
      Download access: =>
      Kill access: =>
      Transfer access: =>
      SysOp access: =>
```

When you press RETURN at one of these prompts, you are leaving that ability at the default for that area. When you type "+" you are adding that ability to the user, overriding the default for the area. When you type "-" you are removing that ability from the user, overriding the default for that area.

DLG File Software

The file areas in DLG are operated through a software module called "DLG:File". If you examine the Main Menu (see the chapter on "DLG Menus") you will find that the "File Base" entry simply calls the "File" module as an executable. You cannot call "File" directly in a CLI, as it requires a certain environment to be present so that it can function correctly. However, it is similar to CLI commands, in that it can take certain command line arguments that will affect how it behaves:

```
DLG:File [-a <area> -c <command stack> -s <sig> -f -p <level> -t -m]
```

Here is an explanation of each argument:

-a <area> : You can create a custom entry for File in your Main Menu that will cause the user to be taken to the given area whenever they enter the file base using that command. Bear in mind though that if the user has private files, this command line argument will have no effect - the user would enter the file base in the file area you specify, but would be immediately redirected to their private file area. The solution to this is to use the **-c <command stack>** argument, discussed below.

-c <command stack> : You can create a custom entry for File in your Main Menu that will pre-pend a custom command stack onto any existing command stack the user might have. This command stack can then direct the user into a particular course of action. If the command stack starts with the "~" character, then any private files the user might have in their private file area will be ignored. Otherwise, DLG would redirect the user to their private file area, and forget the rest of the command stack.

You can also use the command stack argument to make other arguments work properly. Let's say that you wanted to use the **-a <area>** argument to take the user to file area 10 from a Main Menu command. You would create a command line for your new menu entry as follows:

```
DLG:File -a 10 -c ""
```

This would prevent DLG from redirecting the user to their private file area if they had new mail waiting.

-s <signumber> : This command line argument will put a user into a particular SIG when they enter the file area.

-f : This command line argument works with the **-s <signumber>** argument in that it will force a user to remain in the SIG that you specify with the **-s** argument. The user will be unable to select a different SIG.

Using this feature, you can now create several different file area SIGS available from the Main Menu of your BBS. For example, let's say you have created a SIG for Amiga users — SIG number 1. Your Main Menu could have a command "Enter AMIGA File Base", and you could call File with the appropriate command switches when you define the menu item:

```
DLG:File -s 1 -f
```

A user entering the file base in this fashion will not be able to switch SIGS without returning to the Main Menu. This feature allows you to create the illusion of having several special file areas. If you leave the **-f** switch off the command line for File then the user will start out in the indicated SIG as a default, but will be able to change SIGs once they enter the file base. If you call FILE with no switches (as your system is configured by default), the users are defaulted to the last SIG they visited on their previous call.

-t : If you include the **-t** argument in your menu entry for FILE, DLG will stop the user's clock during uploads. This is so that a user's on-line time will not be affected by uploading files. Keep in mind that a user with a time limit of 15 minutes could possibly tie your system up for hours uploading files with this option engaged. SysOps usually implement this feature to give users a bonus for uploading files.

-m <menuname> : This is the name of a configurable menu to use instead of the default built-in menu. Using this option can allow you to add or remove commands from the default File menu.

-p <level> : This indicates a private send level. Either the sender or the receiver has to have a user level that is at or greater than the level specified here. For example, if you set the **-p** option to level of 255, then anyone would be able to send files to the SysOp, and the SysOp would be able to send files to anyone, but none of the user would be able to send files to each other.

Miscellaneous Notes

- Each file in an area has a number which indicates the order that it appears in the file area. Each user in that area has a pointer which indicates the highest file they have seen in that area.
- DLG allows you to create "group accounts." These accounts are collections of users that you can define with the "Group Account" editor in the SysOp menu. Unlike private messages, private files cannot be sent to each member of a group, as this would lead very quickly to multiple copies of the same file being sent to each member of the group, uselessly filling up the hard drive space. It would be much better to create a special access file area and add all of the members of the group to whom you wish to give access to the file or files. If a file is uploaded to a group account, the file will be sent to the GroupOp of that group.
- The batch upload and download operations of DLG are very robust. When users upload files, they can upload as many as they want, providing they are using a protocol that supports it (for example, Zmodem). They will be asked to describe the uploads once they are finished sending files. If for some reason they are unable to do so, (drop of carrier, out of time), they will be prompted to enter the descriptions on their next call.
- Partial uploads are also possible. Users can start an upload using a protocol that allows for resumption of transfer (Zmodem). If the user is disconnected before that upload is finished, the partial file is stored for the next time they log in. At their next visit to the file area, they will be reminded about the partial file and given the option to finish uploading it, or to delete the partially uploaded file. A user should never have to upload a file, or part of a file more than once to the system. DLG is very robust in keeping track of the files received and allows a file to be uploaded over the course of several calls if need be.

DLG Text, Batch and Configuration Files

The previous chapters outlined the steps you needed to perform to install and set up your DLG system. You now have the information and skills required to create and maintain message and file areas. You have also been introduced to some of the underlying concepts in the DLG system.

This chapter will indicate some of the steps you can take to further customize your DLG system. There are a number of text files that you will want to edit to include personal information about your system, establish system rules, provide extra help for your users, etc.

In contrast to previous chapters, this chapter does not contain any tutorials. It contains reference information and general guidelines as to how to proceed.

We assume that you have some familiarity with editing text with any text editor you have available. You may use your favorite word processor to edit DLG text files, as long as it is able to save in a plain ASCII text format, without adding extra linefeeds to the ends of lines (Transwrite, ProWrite, Final Copy, and Excellence! are examples of word processors that can save files in a plain ASCII format). DLG text files should be saved in a paragraph format - DLG will automatically word-wrap the text according to individual user's screen settings when displaying the files. For this reason, we do not recommend using ED, AmigaDOS's small text editing command. ED is more geared towards the creation and editing of batch files, which we will cover in the next chapter. In the event that you do not have a suitable text editor, ED will do in a pinch, but be aware that DLG's word-wrap routines will be undone by the forced formatting in an ED-created file. Users will an 80 column screen will have no problems, but users with shorter screen width may experience some strangely formatted results. You can also log into your DLG System, use the Drop to DOS function from the Utilities Menu, and use DLGEdit to edit text files on your system. DLGEdit will use whatever DLG text editor you have chosen in your user options (by default, ScreenEdit).

%Switches

The first new concept we need to examine is the concept of "%switches". A %switch is a percent character followed immediately by one of a given set of keywords that DLG will recognize. %Switches can be used in DLG text files, menu sets, and language files, so some of them presented here do not apply fully to text files.

When DLG encounters a %switch in a text file as it is being displayed, it will substitute a piece of information for the %switch. There are a number of different switches that DLG provides for, which makes for some very interesting, and personalized text files. Case is not important to %switches. Switches that refer to user data will always report the data from the current user's account.

%Switches can be optionally followed by a period and a number. This will cause DLG to format the substituted text to be exactly as wide as the indicated number. Substitutions that are longer than the switches are cut off, while shorter substitutions are padded out with spaces. This makes it easy to create attractive formatting with variable information. For example:

```
Name: %NAME.26 Date Joined: %JOINED
Address: %ADDRESS.26 Last Login: %LASTCALL
Terminal: %COMPUTER.26 User Level: %LEVEL
```

The .26 appended to the first set of %switches will ensure that the second column of information will all be lined up, even though the substitute text will be of varying length.

Name: James Hastings-Trew	Date Joined: Fri 9 Jul 92 21:06
Address:	Last Login: Sat 10 Jul 92 0:14
Terminal: Amiga	User Level: 255

Here is a complete list of all DLG %switches with an explanation of each one:

User Personal Information Switches:

%NAME	-The full name of the user
%FIRST	-The user's first name
%LAST	-The user's last name
%UNAME	-The user's name with an underscore character (i.e. Kim_Green)
%ADDRESS	-The address of the user
%CITY	-The user's city
%PROVINCE	-The user's State or Province
%COUNTRY	-The user's Country
%POSTAL	-The user's ZIP or Postal Code
%PHONE	-The user's phone number
%BYEAR	-The year of the user's birthday
%BMONTH	-The month of the user's birthday
%BDAY	-The calendar day of the user's birthday
%AGE	-The age of the user

User System Information Switches:

%JOINED	-The date the user joined the system
%LASTCALL	-The date the user last called the system
%CALLS	-The number of calls the user has made to the system
%ALIAS	-The user's alias
%PASSWORD	-The user's password
%COMPUTER	-The type of computer the user has
%LEVEL	-The user's User Level
%SCLNGTH	-The length of the user's screen
%SCWIDTH	-The width of the user's screen
%HELPLVL	-The user's help level - either Novice, Intermediate, or Expert
%PROTOCOL	-The user's chosen download protocol
%UPROTO	-The user's chosen upload protocol
%SYSPAGES	-The number of times the user has paged the SysOp
%DAYLIMIT	-The user's daily time limit in minutes
%TLTODAY	-The number of minutes left in the user's daily limit
%SESLIMIT	-The user's time limit per session
%TLCALL	-The number of minutes left in the user's current session
%TUTODAY	-The number of minutes the user has used today
%TUTOT	-The number of minutes total that the user has been on the system
%DIRLIMIT	-The size of the user's allowed private directory
%MSGENTER	-The number of messages the user has written
%MSGREAD	-The number of messages the user has read
%BYTESUP	-The number of bytes the user has uploaded
%BYTESDN	-The number of bytes the user has downloaded
%FILESUP	-The number of files the user has uploaded
%FILESDN	-The number of files the user has downloaded
%DLBYTES	-The number of bytes the user is allowed to download (-1 if no limit)
%RATIO	-The user's upload/download ratio limit
%LASTMSG	-The number of the last message area the user visited
%LASTFILE	-The number of the last file area the user visited
%PORT	-The three letter name of the port the user is current logged into
%BAUD	-The baud rate of the user's current connection
%CREDIT	-The user's current NetMail credit
%NETPRIV	-The state of the user's NetMail privileges
%UUCP	-The user's UUCP status
%ANSI	-The user's ANSI status (either "color" or "mono")

%MENUSET -The user's selected menu set

General Information Switches:

%DATE -The current date and time

Display Control Switches:

(NOTE: unlike the variable %switches above, these %switches MUST appear at the START of a line. If there is a numeric parameter required, be sure to follow the parameter with at least one space.)

%MOREOFF	-Disable "More" prompts
%MOREON	-Restore "More" prompts to user's settings
%WRAPOFF	-Disable word-wrap
%WRAPON	-Restore word-wrap
%POSOFF	-Disable ANSI screen-positioning
%POSON	-Restore ANSI screen-positioning to user's settings
%CLROFF	-Disable screen clears
%CLRON	-Restore screen clears to user's settings
%COLOUROFF	-Disable ANSI colour
%COLOURON	-Restore ANSI colour to user's settings
%RETURN	-Display "Press RETURN" prompt
%DOMORE	-Force a "More" prompt
%INDENT.<x>	-Indent text by <x> number of spaces
%SETWIDTH.<x>	-Set the user's screen width to <x> for current file
%PAUSE.<x>	-Pause for <x> seconds
%SLOW.<x>	-Insert <x> number of ticks (50th of a second) between characters
%BREAKON	-Enable CTRL-C breaking
%BREAKOFF	-Disable CTRL-C breaking

ANSI Control Switches

%a<n>	-Change the following text to colour <n> (0 to 7)
%ab	-Change the following text to bold
%ai	-Change the following text to italic
%au	-Change the following text to underlined
%ar	-Change the following text to reverse video
%af	-Cause the following text to blink (not supported in most Amiga terminal programs)
%ao	-Reset all text modes to normal

DLG Text Files

Here is a list of the various text files that DLG will locate and display when a user moves about your system. You will want to edit most of these to personalize your DLG system. A brief description of the contents of each text file, and when the file is displayed by DLG follows. The files are presented in groups according to how important they are for you to modify. The file names are shown as the complete pathname that DLG will look for the file in. In the case of text files that should be kept in file or message area directories, the pathname will be given as VOL:[areanumber]/filename.txt, where VOL: would be either FILE: or MSG: and [areanumber] would be the number of the directory in which the file should be located. For example, to put an EnterArea.txt file in message area 10, its pathname would be MSG:10/EnterArea.txt

The following text files must be changed. The DLG installation program placed generic versions of these files, but since they are highly visible (seen each time a user is on your system) it is advisable to change them right away.

DLGConfig:Text/Title.txt

This text file is shown before a user logs on, right after a connection is made into the DLG system. This file is a default, which can be replaced by the following text file on a port by port basis.

DLGConfig:Text/Title.[port]

This is a special title text file that is displayed for one particular port. The [port] would be the three letter name that DLG identifies the port by. For example, to put a different title file on port TR1:, you would have a text file called DLGConfig:Text/Title.TR1. This text file overrides the global DLGConfig:Text/Title.txt.

DLGConfig:Text/NewUser.txt

DLG displays this to someone logging on as a new user, before they fill out the new user application. This is a good place to describe your system to the user, and inform them of any special rules, fee requirements, etc.

DLGConfig:Text/Application.txt

The DLGConfig:Text/Application.txt file is a special case text file. It can contain verbose explanations of each question, as well as the questions themselves. It also has a very particular structure and order which must not be altered.

There are special %switches used in the application.txt file to modify how the DLG system uses it while a user is filling out the application.

%q - This special %switch designates the line as a question. Text that describes what the question is about should precede the question line. When the question line is reached, DLG displays the question with an appropriate prompt. Since the questions come in a very particular order, DLG knows what to do with the response to the question, and form a prompt that is appropriate to the question.

%t - This special %switch indicates that the line is a template. It precedes a question line, and forms a template for input to the question. The template can be any combination of text characters interspersed with underscore characters. The underscore characters form the input areas of the template. DLG will print the other characters as the template is filled out.

Here is an example template:

```
%t(____) ____- ____
```

```
%qWhat is your phone number? —>
```

As the user enters their phone number, DLG will automatically provide the “()-” characters as they are required. A templated question requires full input — i.e. all template spaces must be filled out.

%s - This special %switch indicates that the line is a question, but one that is to be skipped. For example, if you were setting up a system for a group of very inexperienced computer operators, you may wish to skip most of the questions regarding file transfer protocols, editing options, and so on. DLG will fill in skipped questions with built-in defaults.

DLGConfig:Text/FinishedApplication.txt

This text file is shown when a user has completed the new user application, but before the user logs onto the BBS.

DLGConfig:Text/Login.txt

This text file is shown when a user logs on, right after the execution of the DLGConfig:Batch/Login.DLGBatch and/or DLGConfig:Batch/Login.Batch files.

DLGConfig:Text/Logout.txt

This text file is shown when a user logs off the BBS.

The following text files can be changed, but it is not crucial to do so. You may find that the DLG default text files for many of these items provide exactly the right information and tone that you wish to convey.

DLGConfig:Text/2MinWarn.txt

DLG displays this file when the user has 2 minutes left in the current session.

DLGConfig:Text/0MinWarn.txt

DLG displays this file when the user has run out of time on the system. It is displayed just before the user is automatically logged out of the system.

DLGConfig:Text/Event.txt

DLG displays this to inform the user that their time for this call has been shortened due to an event.

DLGConfig:Text/HappyBirthday.txt

This text file is shown when a user logs on and it is their birthday.

DLGConfig:Text/NoProceed.txt

DLG displays this file when a user logs on and there is not enough time before an impending mail event to allow them to do anything worthwhile.

DLGConfig:Text/NoRegistration.txt

When users who are not recognized by DLG log into the system they are given the opportunity of filling out an application form. If they choose not to fill one out, this text file is displayed. The intention here is to inform the user of what is happening, in case they made an error in typing their name, or there was line noise, when they first logged in.

DLGConfig:Text/NotEnoughTime.txt

This text file is very similar to NoProceed.txt, in that it is displayed to users who have less than three minutes left in their daily time limits. The text is displayed, and the user is logged out.

DLGConfig:Text/RefuseDownload.txt

If you have imposed file upload/download ratios on your users, then users who have downloaded too much will see this text file when they attempt to download a file that would exceed their ratio.

Optional Text Files

The following text files are entirely optional. If they exist, DLG will use them.

DLGConfig:Text/Echo.txt or MSG:[AreaNumber]/Echo.txt

This file, if in an echomail message directory, will display just before a message is entered in that directory. If there is no echo.txt file in the current message directory, the default DLGConfig:Text/Echo.txt will be shown instead. So, if a user were about to enter a message in Message area 10, and there existed a file called Msg:10/Echo.txt, that file would be shown. This is a good mechanism to remind users about echo rules, and ensure they stay on-topic.

DLGConfig:Text/UploadFile.txt or File:[AreaNumber]/UploadFile.txt

If DLGConfig:Text/UploadFile.txt exists it will be shown to a user just before they upload a file. If for some reason you want a particular area to show a different file, you may place one in that area's directory. It will override the one in DLGConfig:Text/.

Msg:[AreaNumber]/EnterArea.txt or File:[AreaNumber]/EnterArea.txt

If this file exists in any given message or file area, it will be displayed when a user enters that area. Use this for areas with special contents or rules of conduct.

DLGConfig:Misc/Screen.MSG or FILE:[Areanumber]/Screen.MSG or MSG:[Areanumber]/Screen.MSG

This is a text file containing pairs of words like so: "Icat!" "Ifeline!". This text file is used to create Screen.DAT files, which can be used to filter out unwanted language in your message and file areas. When the "I" appears at the beginning of a word in the Screen.MSG file, it means that "This is the beginning of the word" and that no other letters should appear before this string of characters. When the "I" appears at the end of a word in the Screen.MSG file, it means that "This is the end of the word" and that no other letters should appear after this string of characters. When the entire string is bracketed by "I" characters, this means that the word is to be taken as an entire word, all by itself.

Once you have created your Screen.MSG file, you need to CD to where the Screen.MSG file is located, and run the CompileScreen program to compile it into a format that DLG can use.

DLGConfig:Misc/Screen.DAT will be the global screening file for all your message and file areas. If you have special needs in particular message or file areas, you will need to create separate Screen.DAT files in those area directories. The area specific Screen.DAT file will over-ride the main Screen.DAT file. Note: only messages that have been entered locally are filtered. Messages that arrive on your system in EchoMail, NetMail, or UseNet areas are unfiltered.

DLGConfig:Text/BaudRateTooLow.TXT

This file will be shown to any user who logs into a port on your system with a baud rate that is lower than the minimum baud rate that you have defined for that port.

DLGConfig:Text/PortIsPrivate.TXT

This file will be shown to any user who logs into a port on your system that has been designated to be a private port, and that user is not a member of the group that has been defined as that port's access group.

The following text files are accessed only by items in the various DLG menus. Change them if you need to, or ignore them if you want.

DLGConfig:Text/BBS-Manual.txt

This text file is called from the default Utility menu, and is shown using the DLG:DF command. It is an on-line manual for the use of DLG. You may want to personalize this file if you make any heavy alterations to the layout of your system. This manual only reflects the default DLG system.

DLGConfig:Misc/ScreenEdit.help

This text file is shown when a user is composing a message using the Full Screen Editor, and uses the ESC-? or ^K-? sequence for help within the editor.

DLGConfig:Text/MyQuotes.quote

DLG's Quote program uses this file to show various quotations. Here is the structure of the quote file:

```
Quote 1
%%
Quote 2
%%
Quote 3
%%
```

A quote may consist of a number of lines, and may contain blank lines. The %% is the delimiter that shows the end of one quote and the beginning of the next. This file can be edited to add and delete quotes to match the personality of your system.

DLGConfig:Text/Today.<month>

There is a separate Today.<month> file for each of the twelve months of the year. These text files contain all events that will be shown by DLG's **Today** utility. The format of a line in the Today.<month> files is as follows:

```
XMDDDDYYYY Text-of-event
```

MM = month; DD = day; YYYY = year. X can be one of two letters: B for Birthday, S for a special event. (Note: the "B" and "S" designations are not currently supported, but should be included.)

Events may require more than one line. Such events would look like so:

```
XMDDDDYYYY First-Line-Of-Text
XMDDDDYYCYCSecond-Line-Of-Text
```

where the dates are identical. C indicates that the second line is a continuation of the first. The date is only displayed once.

The YEAR can also be left blank, to indicate an annual event like the Chinese New Year.

```
S0204 Chinese New Year
```

With this information in hand, you can add events that pertain more towards your area, country, or user base.

DLGConfig:Text/UserInfo.txt

This text file is displayed from the Utility menu using DLG:DF. It contains text interspersed with DLG %switches which show information about the user.

DLGConfig:Text/LastBoot.txt

You cannot modify this text file, as it is automatically generated by the DLG.Startup script file each time you restart your system.

DLG Language files

All of the strings in DLG are contained within language files. A language is a text file with many imbedded %switches to indicate various things like data type, ANSI colour, text replacements, and so on. (Note: the %switches used in the language files are NOT the same as the DLG %switches used in text files — see below for details). It can be very dangerous to tamper with some of the strings in the language files. Please use the following guidelines when modifying the strings to create alternate language files:

- 1 Never, under any circumstances, edit the 'English.lang' file. If you wish to make changes, duplicate this file under a different name and modify the duplicate. Any port can then be told to use this new file by modifying the global configuration for that port.
- 2 Each string is enclosed within quotation marks and this completely delimits the string. There is one string contained on each line of the language file.

- 3 Any string that is not a recognizable string from the BBS should not be changed.
- 4 Any string with an '=' after the string should not be changed. ie. "this is the string"=
- 5 No lines may be inserted or deleted from this file. Strings are position sensitive in this file and any additions or deletions will cause a string offset to occur thus making the DLG software crash.
- 6 Any lines containing a numeric value after the string should be no longer than the numeric value indicated. For example:

"this is the string",25

The above string can be modified, but cannot be made longer than 25 characters in length. This length does not include the quotation marks. Do not try to make the string longer by changing the number — that will not work.

- 7 The special %switches you see in the strings are partially ANSI colour, and partially C variable substitution switches. Here is a brief explanation of the switches
 %s - A string will be substituted for this code. **DO NOT CHANGE**
 %d - A number will be substituted for this code. **DO NOT CHANGE**
 %ld - A number will be substituted for this code. **DO NOT CHANGE**
 %a<0-7> This is an ANSI Colour switch and can be changed or modified as you see fit. This will set the text following the ANSI switch to corresponding ANSI colour. Please avoid setting the text to colour 0 or colour 4. Colour 0 is usually the colour of the screen background, and colour 4 will be the same as colour 0 on a standard four colour Amiga console or terminal program screen.

The system will presently default to English lang. This setting can be changed for any port by modifying the Global set for that port.

Once a new language file has been created, it must be compiled with the CompileLang utility before it can be used by DLG. Once this is done, you can attach your new language file to a menu set by defining a new menu set for this language. Note that if you create a new language file, you should (but don't have to) also create a separate set of menus in that language as well — the text used in DLG menus are part of the menu sets, NOT part of the language files.

Character Translation Sets

A major part of DLG that will be of use to European systems is the ability to re-map any input character to an alternate character, and to re-map any output character to an alternate character. An example would be taking foreign language characters (High Bit Characters), translating them to something allowed by the FidoNet 7 bit ISO-LATIN 1 character set, storing the message, and translating them back the other way when they are transmitted.

Most North American systems will likely not need to worry about this, and will not need to make any changes.

There is a directory in DLGConfig: called 'Charsets'. Within this directory are two types of files, a master control file named CharSets.bbs and the input/output translation files named <name>.set.

The SysOp can add, delete, view, or edit the translation files using the General Configuration Editor from the SysOp Menu. The following information is stored about each character set:

- 1 Number of set. (This is the number that the system/user will use to refer to the character set.)
- 2 Name of the character set. This is the descriptive name of the character set. The name of the character set is also used to determine its data file name (ie <name>.set).
- 3 A user level, determining which users are able to choose the character set.

4-7 These are the actual character translations. 4 and 5 deal with the input translations and 6 and 7 deal with the output translations.

You may set up a default character translation for each port, and users may also select alternate character set maps through their user options or when they apply for an account.

- 1 In the globals configuration, a default translation can be specified.
- 2 A user may override this setting by choosing a different character set in the User Options module. If you create a useful character translation, we would appreciate a copy of it along with a description of what character set it is, and what country it is generally used in.

Script Files

There are two basic types of "script" or "batch" files used with DLG. Script files are lists of commands to be executed, and are identical to the type of script file that AmigaDOS uses. For example, your Amiga's "Startup-Sequence" is a script file, and contains a listing of all the commands your Amiga needs to execute when it is powered up. The "DLG.Startup" file is another example of this type of script file. A script file can contain any type of AmigaDOS or DLG command, and present information to the user, but a script file cannot receive input from a user.

The other type of script file used with DLG are DLGBatch files. These look nearly identical to normal AmigaDOS script files, but there are four differences. DLGBatch files cannot contain any IF, ELSE, ENDIF, SKIP or LOOP statements - (i.e. they must be linear script files with no decision making or branching operations). The second difference is that DLGBatch files can use %switches. The third difference is that executables run in DLGBatch files can accept input from users, while normal batch files cannot. The fourth difference is that DLGBatch files cannot be passed arguments.

Here is a listing and explanation of all of the script files used with DLG. They are organized into two parts - essential scripts necessary to the operation of DLG, and optional scripts that will be called into play if they exist.

Essential Script Files

DLGConfig:Batch/Chat

This script file is called from the Main Menu [C] Chat command. The script name is not hard-coded, and can be called anything you like. Since the chat command is only called by this script file, it is possible to replace DLG:Chat with any other third party chat program.

DLGConfig:Batch/TPTScript

This file is used if you wish to run TPTCron in its own window. It calls and initializes TPTCron with the DLGConfig:Batch/CronTab config file. Our installation program directs the output of TPTCron to NULL: instead of using this file.

DLGConfig:Batch/TLx.startup

TL0.startup, TL1.startup, etc, are the batch files executed whenever a user logs on to the corresponding local port.

DLGConfig:Batch/TRx.startup

TR0.startup, TR1.startup, etc, are the batch files executed whenever a user logs on to the corresponding port. These are remote connections via the appropriate serial port.

DLGConfig:Batch/UUCP.batch

This script file is executed whenever a user who has been defined as a UUCP client logs in. This script is executed and the call is terminated. This script file is usually used to run GETTY.

S:DLG.Start

This script file starts up the DLG system. It makes assignments, starts the TPTCron and TPTRM (resource manager) program modules, and activates the ports. This script file can be executed manually each time you start up your system, or you may have it execute automatically from your own S:Startup-Sequence (WB 1.3), or S:User-Startup (WB 2.0).

S:Local

This script file is used to initiate a local session.

Optional Script Files

DLGConfig:Batch/AppliedTemplate.batch

This script file is called by the SysOp User Editor when you alter a user's account by applying a template. This optional batch file is provided for you to do other modifications to the user's account not provided for by the SysOp User Editor's template function.

The script file is called in the with the following parameters:

.key first_last/a, OldLevel/a, NewLevel/a

where First_Last, OldLevel, and NewLevel are passed to the script file.

DLGConfig:Batch/Logout.DLGBatch or DLGConfig:Batch/Logout.Batch

Once a user has elected to log out of the system, this optional batch file is executed. This is run just before the optional Logout.txt file is displayed.

DLGConfig:Login.DLGBatch and/or DLGConfig:Login.batch

Once a user has logged in, DLG will execute either Login.batch and/or Login.DLGBatch. This script file is normally used to display system information and check for waiting mail by use of the WMail program. If used as a DLGBatch file, this script will be able to use the %switches and accept input from the user if required. It is preferable to call this as a DLGBatch file if you are going to use the WMail program, as the %switches make it easy to pass command line parameters to that program.

DLGConfig:Batch/LocExtEditor

This script file will allow the use of external text editors or word processors with DLG. This capability is only for the local port, as this script gives you the ability to use any Intuition-based software for the editing and creation of messages. Intuition-based software cannot be used on a remote port. The example script (called .LocExtEditor to disable it) shows how to use WordPerfect as the local text editor with DLG. Any software can be used provided it is able to save in a plain ASCII text format, and that it does not detach from the CLI process it is called from. An example of a text editor that would not be suitable would be Gold Disk's TransWrite, as this program detaches from the CLI.

DLGConfig:Batch/NewUser.batch and/or NewUser.DLGBatch

This optional script is executed once a user has completed the new user application. It is provided so that you can add features such as callback verifiers, or other questionnaires to your system.

DLGConfig:Batch/ReceivedFile.batch

DLG will execute this script file, if it exists, after each successful file or batch upload. It is called with the following parameters:

.key PathFileName/a, FileName/a

where PathFileName and FileName are passed to the script file. PathFileName is the path/filename of the file, and FileName is the filename of the file, minus the path.

DLGConfig:Batch/Spare(n)

In the early days of DLG, the message translation function required the use of one of three named script files - Spare1, Spare2, and Spare3. The message translation function now allows for the use of any named script file, not just one of these three presets. These script files are still valid examples of how the message translation function works. Spare1 uses the Kraut program to add a German accent to English text; Spare2 uses the Jive program to add an American dialect accent to English text; and Spare3 uses the Valspeak program to add a different American dialect accent to English text. You may rename these script files to whatever you like, as long as you remember to change any message area setups that use them.

DLGConfig:Batch/SystemInfo.batch

Traditionally, this is the batch file executed from the Utilities menu when a user wishes to view system statistics. It usually runs programs such as 'info' and 'status'. There is no requirement that this actual name be used, as long as the menu is configured to call the proper name.

DLGConfig:Batch/Transfer.batch

This batch file is executed just before a file is transferred or validated.

DLGConfig:Batch/Upload1.batch

This batch file is executed just before a user enters a file description for an uploaded file. It is called with the following parameters:

.key UserName/a, PathFileName/a

where UserName and PathFilename are passed to the batch file. UserName is the underscored name of the user who uploaded the file, and PathFileName is the path/filename of the uploaded file.

DLGConfig:Batch/Upload2.batch

This batch file is executed just after a user enters a file description for an uploaded file. It is passed the following parameters:

.key UserName/a, PathFileName/a

where UserName and PathFilename are passed to the batch file. UserName is the underscored name of the user who uploaded the file, and PathFileName is the path/filename of the uploaded file.

DLGConfig:Batch/ValidatedUser.batch

This batch file is called when a user is validated, with the following parameters:

.key UserName/a, OldLevel/a, NewLevel/a

where UserName, OldLevel, and NewLevel are passed to the batch file. UserName is the underscored name of the user being validated, OldLevel is the old user-level before validation, and NewLevel is the new user-level after validation.

DLGConfig:Batch/ViewArchive.batch

This batch file is called when a user wishes to view the contents of an archived file. It simply calls DLG:ViewArchive, passing the filename.

DLGConfig:Batch/DrivesFull.Batch

This batch file is called when an upload cannot be completed because the volume onto which the file is to be moved is full. It is called with the following parameters:

.key SourceName/a, DestName/a

where SourceName and DestName are passed to the batch file. SourceName is the source pathname of the file, and DestName is the destination pathname. You can use the SendMsg command (see the reference chapter) to send yourself a note that the hard drive is full by using this batch file.

DLGConfig:Batch/IB

This batch file is called when DLG detects a serious problem with the software. If you ever see this file execute, please contact TelePro Technologies immediately and let us know that this has happened. Under normal circumstances, you will never see this file execute.

FidoNet Script Files

The script files listed below are used when you interface your DLG system with a FidoNet network. These script files are used by a variety of programs - DLG, DLGMail, and TrapDoor. Please read the chapters on DLGMail and TrapDoor before attempting to modify these script files. Note that the script files provided are merely examples - they *must* be edited as part of the process of setting up your FidoNet system.

S:FidoStart.batch

Similar in nature to DLG.Startup, this script file sets up assignments and variables for DLGMail. It also starts the background shell that DLGMail requires. You should probably execute this script from your DLG.Startup script if you are using DLGMail on your system.

DLGConfig:Batch/Crashmail.batch

This script is executed by DLG when any NetMail message is entered with the CRASHMAIL option set. This script will need to be properly configured for your particular FidoNet setup.

DLGConfig:Batch/KiITD

Other FidoNet script files use this script file to disable the TrapDoor program. This must be configured properly for your particular FidoNet setup. Note that if you are using DLGMail, this script file should not be necessary.

Fido:batch/MakeAlias.Batch

This is a DLGMail script which is executed when the DLG mail processor starts up. It establishes various AmigaDOS shell aliases as shortcuts. For example: Alias die DMC TDOFF

Fido:batch/Maintenance.batch

This is a DLGMail script, and is usually executed on a timed basis by TPTCron. It runs various utilities to freshen files, generate file lists, purge user directories, trim log files, etc.

Fido:Batch/NLComp.DMB

This is a DLGMail script file, and is executed by DMC to compile a nodelist.

Fido:Batch/NLDiff.DMB

This DLGMail script file is executed by DMC to compile a new nodelist when a NodeDiff has been received.

Fido:Batch/NLNew.DMB

This DLGMail script file will compile a newly received nodelist. It is usually executed by DMC. It should not be run automatically, but can be.

Fido:Batch/Post-Dial.DMB

This DLGMail script file will be executed after DLGMail has completed all dialing out.

Fido:Batch/Post-DLGBundle.DMB

This DLGMail script file will be executed after DLGMail bundles mail packets.

Fido:Batch/Post-DLGNet.DMB

This DLGMail script file will be executed after DLGMail processes NetMail.

Fido:Batch/Post-Import.DMB

This DLGMail script file will be executed after DLGMail imports mail.

Fido:Batch/Post-Export.DMB

This DLGMail script file will be executed after DLGMail exports mail.

Fido:Batch/Post-Tick.DMB

This DLGMail script file will be executed after processing *.TIC files, but ONLY if TICK is enabled by DLGMail.

Fido:Batch/Pre-Dial.DMB

This DLGMail batch file will be executed if DLGMail has determined that a call will be made, BUT before dialing.

Fido:Batch/Pre-DLGBundle.DMB

This DLGMail batch file will be executed before DLGMail bundles mail packets.

Fido:Batch/Pre-DLGNet.DMB

This DLGMail batch file will be executed before DLGMail processes NetMail.

Fido:Batch/Pre-Import.DMB

This DLGMail batch file will be executed before DLGMail imports mail.

Fido:Batch/Pre-Export.DMB

This DLGMail batch file will be executed before DLGMail exports mail.

Fido:Batch/Pre-Tick.DMB

This DLGMail batch file will be executed before processing *.TIC files, but ONLY if TICK is enabled by DLGMail.

Fido:batch/Tick.DMB

This DLGMail batch file will be executed when DLGMail detects *.TIC files in the inbound directory AND if it is enabled to run in DLGMail.cfg.

Fido:Batch/ZMHOff.DMB

This DLGMail batch file will be executed after DLGMail has set TrapDoor to normal BBS operation, as opposed to ZMH (Zone Mail Hour) status.

Fido:Batch/ZMHOn.DMB

This DLGMail batch file will be executed after DLGMail has set TrapDoor to ZMH (Zone Mail Hour) status. During ZMH, TrapDoor will not accept any connections from human callers — only calls from other FidoNet systems will be handled.

DLG Configuration Files

This section covers reference material on the various configuration files used in the DLG system. These configurations files are simple text files, which you can change using any text editor. The AmigaDOS "ED" command is a good tool for editing these types of file. What follows is a listing of all of the various configuration files used by DLG and related products, with a brief description of what each one is used for.

DLGConfig:Batch/CronTab

The crontab is a text file used by TPTCron. This file lists all of the timed events that TPTCron manages. For a detailed explanation of the format for the CronTab file or the TPTCron program please see the chapter on DLG Commands.

Fido:DLGMail.ARE

This DLGMail config file is used to establish what EchoMail areas are carried, and correlates the tagnames with feeds and message areas. It also controls other aspects, such as AreaFix classes, multizones, passthrough areas, and linking.

Fido:DLGMail.BUN

This DLGMail config file is used to establish the archiver methods and calling styles used with certain nodes.

Fido:DLGMail.CFG

This is the main DLGMail configuration file and establishes most operational options for DLGMail.

Fido:DLGMail.MAC

This DLGMail config file is used to set certain macros to be substituted for long DLGMail command lines.

Fido:DLGMail.MRT

This DLGMail config file is used to establish mail routing to other nodes.

Fido:DLGMail.PNT

This DLGMail config file is used in re-routing NetMail addressed to your point operators.

Fido:PDQMail.AFX

This PDQMail config file is used to establish AreaFix passwords, echo classifications, and other relevant data to be used with PDQAreafix and PDQAreafix+.

MSG:<n>/Screen.msg or FILE:<n>/Screen.msg or DLGConfig:Misc/Screen.msg

Message screening is the process by which words or phrases contained in messages and file descriptions are removed and are replaced with more suitable substitutes. A Screen.msg file is a text file outlining the relationships between the original words or phrases, and their substitutes. The search process is case insensitive - when the original word begins with a capital letter, the substitute word is also capitalized. Words and their substitutes are listed in pairs, enclosed in quotation marks. A special wildcard character ("I", found between the + and backspace key on the Amiga keyboard) is

used to match any non-text character. In this way, it can be used to mark the start and end of words, as a wildcard for spaces and punctuation marks. For example, if you wanted to replace all occurrences of "cat" with "feline", you would configure your screen.msg file to include the line:

```
"|cat|" "|feline|"
```

If you did not delimit the word cat as in this example, then the screen process would result in the word "catatonic" being translated to "felineatonic", which probably isn't what you intend to happen.

Screen files are located in one of three places - in the message area they are intended for, in the file area they are intended for, or in DLGConfig:Misc, where it will be used as a global filter for all message and file areas. Before the Screen.msg file can be used by DLG, it needs to be compiled into a screen.dat file using the CompileScreen program. The process is outlined in the chapter on DLG Command files.

DLGConfig:Misc/TPTFreq.lst

A "file-request" is a FidoNet convention whereby one system can request that another system send it files from its file areas. The files that are available for this type of transfer are listed in a text file, usually called FILES.TXT. When you are running a file-request aware front-end program, like TrapDoor, a system can call in and make a file request for this list by using the "magic" filename FILES. Magic filenames can be defined for other commonly requested files on your system. A magic filename is a convenient short-form or mnemonic name that you can define for any file on your system. MakeFList generates a list of files which may be file requested from your system. It creates a text file containing the filenames of files that are not password protected. If DLGConfig:Misc/TPTFreq.LST contains a list of magic filenames, the filenames will be listed as well. This file contains all of the 'magic' names that TPTFreq can recognize, and the actual path to the files that have magic names. Even if your system does not support a large number of magic filenames, it is always a good idea to configure FILES and FILES.TXT, as well as ALLFILES to point to a file list generated by MakeFList. Here is a sample TPTFreq.list file:

```
FILES file:files.txt
:This is a listing of all files requestable from this system.

ALLFILES file:files.txt
:This is an alternate name for the FILES magic filename.

FILES.TXT file:files.txt
:This is an alternate name for the FILES magic filename.

SECRET Drive:filename !password

DLGFEATURES dlgconfig:text/DLGFeat.lzh
:This is a list of features for DLG Professional

FIDONET file:8/FidoNet.lzh
:Documentation about Fidonet, Glossary of Fidonet terms

TRAPDOOR file:8/TD_1_80.LZH
:Latest version of the TrapDoor mailer

TRAPTOSS file:8/TT_1_20.LZH
:Latest version of the TrapToss mail tosser/packer
```

Each entry consists of two lines. The first entry on the first line is the magic filename for a file. The second entry on the line is the actual path and filename of the file that the magic filename stands for. An optional third entry is an exclamation point followed by a password. A file request attempting to get such a file will have to supply the password or the request will fail. The second line of an entry in the TPTFreq.LST consists of a colon, followed by a brief description of the file. Note that password protected files do not need a description, as they will not be listed in the FILES.TXT file.

DLGConfig:Misc/TPTShell.cfg

This file defines what commands are available from the TPTShell (Drop to DOS from the default Utility menu), what commands are executed in their place, and what user levels are required for each. Here is a sample TPTShell.cfg file:

```
Assign c:assign 1 0
List c:list 1 0
Cd c:cd 1 1
Date c:date 1 0
Version c:version 1 0
Path c:path 1 0
Dir c:dir 1 0
Why c:why 1 0
Echo c:echo 1 1
Stack c:stack 1 0
Status c:status 1 0
Avail c:avail 1 0
Info c:info 1 0
```

The first entry in each line is the name of the command that the user can type. The second entry is the name of the actual command to be executed. The third entry is the minimum user-level required to use this command, and the fourth entry is the number of arguments allowed to be passed to the command. You must remember to restrict argument passing, especially in the case of the DIR command, which has an interactive mode that allows the user to do practically anything on your hard drive. The example shown above is a good starting point.

Mail:TrapDoor.cfg

TrapDoor.cfg sets up TrapDoor to work with your mailer and the BBS. It is not required to be in any particular place, but if you do not specify to TrapDoor where its .cfg file is, it will look in Mail:. Mail: is normally assigned to the same directory as MSG:.

Mail:TrapList.cfg

TrapDoor.cfg tells Traplist how to configure your nodelist. It is not required to be in any particular place, but if you do not specify to TrapDoor where its .cfg file is, it will look in Mail:. Mail is normally assigned to the same directory as MSG:.

DLG Menus

This chapter covers the configuration and customization of DLG's menus. DLG has default menus built into all of its modules. These defaults present a consistent interface to users as they move from one part of DLG to another. You may, however, want to modify or replace these defaults with menus of your own creation. DLG has two different means of changing the default menus. The first is called a "Custom Menu Set" and the second is called a "Configurable Menu."

The "Custom Menu Set" is a method whereby you can change the look, but not the functionality, of a menu. A custom menu set is basically a text file, with special %switches, that is displayed in place of the default DLG menu. You cannot add commands with a custom menu set, nor can you change the keys or letters required to use those commands. You can only replace the default text and display with one of your own creation. You can have many different custom menu sets to choose from on your system, and your users will be free to pick whichever one they like. Language choice is also tied to the choice of the custom menu set.

The second method of menu modification is the "Configurable Menu". With the configurable menu you are able to change the command structure of your various menus. You can add or remove commands, and change the keys associated with those commands. The configurable menus are global - they provide the basis for the the custom menu sets that you can create.

This chapter provides you with a tutorial for working with custom menus, and a reference section. In addition to the material in this chapter, you will also have to make reference to the material in the chapter on the DLG executables, if you start to drastically modify the menus in your particular setup.

DLG Custom Menus

As mentioned above, a custom menu is a text file that DLG will display instead of the default menu format. A custom menu set is a complete set of these text files - one for each menu on your system. The custom menu files have a particular structure which we will cover in the following tutorial.

All custom menu files are kept in the DLGConfig:Menu directory. Each custom menu set that you create must be entered into the SysOp General Configuration module. By doing so you create a list of alternative menu displays that users can choose from. The custom menu set files have to conform to the following naming convention:

MenuName.LEV.SET

LEV stands for one of two possible help levels:

NOV (Novice)

INT (Intermediate)

SET stands for the number of the menu set. As you define menu sets in the SysOp General Configuration module, DLG assigns a number to each menu set. You must name the text files associated with a particular menu set to include this number as part of the name. In this way, DLG will know which file goes with which menu set.

MenuName = .menu file name. For each menu in the DLG system, there exists a "MenuName.menu" file that fully describes the contents of that menu. For instance, the Main Menu has a "Main.menu" file in DLGConfig:Menu. A custom menu file for the Main Menu from set number 1, at the novice level, would be called MAIN.NOV.1.

The default DLG menu format is menu set number 0. If you created text files with a SET number of 0, these would replace the DLG default.

You may also create only a partial set of menu files if you desire. DLG will use the menu from a given set, if it exists, or it will display the default set number 0 if there is no custom menu file for any given menu on the system.

Custom menu files can use all available %switches as outlined in the chapter on DLG text files.

Format of a Custom Menu Text File

A DLG custom menu file is a bit of a cross between an ANSI text file and a batch file. You can create ANSI artwork using any of the many ANSI editor available in the public domain, and add the appropriate menu formatting to the file afterwards. Alternatively, you can also create a DLG menu in a simple text editor. It is preferable to use DLG's %switches to set colour and other ANSI settings instead of the hard-coded ANSI sequences that ANSI editors produce. The reason for this is that the %switches will honor the user's ANSI settings, while the hard-coded ones will be sent regardless. The system is set up to be totally open-ended and flexible. Because of the requirements of the menu file interpreter, the final look of the custom menu will be hard to visualize in the text editor, but a little experimentation will bring good results.

A menu entry in a custom menu file consists of a special %switch and two text strings. The %switch indicates the letter of the command in the menu that this text is associated with. The first text string is what will be shown if the user has access to that menu item. The second text string is what will be shown if the user does not have access to that command. The text strings are delimited by curly bracket "{}" characters, to indicate the start and end of each string. If you want a particular string to be blank or empty, use two curly brackets with no character between them.

For example, a menu entry could look like so:

```
%f{[F] File Base }{[F] File Base (unavailable)}
```

or

```
%m{[M] Message Base}{}
```

In the first example, users would see the command "[F] File Base" if they had access to that command, or "[F] File Base (unavailable)" if they did not. In the second example, users would see "[M] Message Base" if they had access, or nothing at all if they had no access.

Here is an example Custom Menu (MAIN.NOV.1):

```
+-----+
| My BBS M A I N M E N U |
+-----+
%m{[M] Message Base }{}%f{ File Base [F]}{}
%u{[U] Utilities }{}%o{ User Options [O]}{}
%p{[P] PeopleTalk Conferencing }{}%s{ SysOp Menu [S]}{}
%c{[C] Chat with SysOp }{}%g{ Goodbye [G]}{}

What is your choice, %FIRST?
```

This menu would appear like so to the average user:

```
+-----+
| My BBS M A I N M E N U |
+-----+
[M] Message Base File Base [F]
[U] Utilities User Options [O]
[P] PeopleTalk Conferencing
[C] Chat with SysOp Goodbye [G]

What is your choice, Gerry?
```


In this example, notice that the formatting of the actual text file differs radically from the actual menu as displayed by DLG. The %switches are replaced by the current user's information, and menu items are displayed only if the user has access to them. However, any text not preceded by a command %switch will be printed verbatim. You can include ANSI sequences and %a %switches in your custom menu sets, as well as regular ASCII text.

Tutorial: Creating a Custom Menu

In this tutorial we are going to create a custom menu to replace the Main Menu of your system. Load up your favorite text editor, and type the following in. For the purpose of this tutorial, do not worry that the commands in this example menu do not match those available in your real Main Menu. When you create your own custom menu files, you will need to make sure that you remember to include all of the commands available in the menu you are replacing:

```
=====
M A I N M E N U - please choose an option:
=====
M - Enter message base F - Enter file base
U - Switch to utility menu S - Switch to SysOp menu
O - Change Your User Options G - Logout - end this call
! - Send feedback to the SysOp
```

Consider this to be the "skeleton" of your new menu. It is easier to format and visualize the new menu if you start by leaving out the special %switches and alternative text.

Now edit this menu by adding opening and closing curly brackets at the start and end of each line with a command listing on it:

```
=====
M A I N M E N U - please choose an option:
=====
{M - Enter message base }{F - Enter file base}
{U - Switch to utility menu }{S - Switch to SysOp menu}
{O - Change Your User Options }{G - Logout - end this call}
{! - Send feedback to the SysOp }
```

Now, copy each command and paste it right after the original, so that each command is doubled. This will help us to maintain our formatting when creating the alternative command displays:

```
=====
M A I N M E N U - please choose an option:
=====
{M - Enter message base }{M - Enter message base }{F - Enter file base}{F - Enter
file base}
{U - Switch to utility menu }{U - Switch to utility menu }{S - Switch to SysOp
menu}{S - Switch to SysOp menu}
{O - Change Your User Options }{O - Change Your User Options }{G - Logout - end
this call}{G - Logout - end this call}
{! - Send feedback to the SysOp }{! - Send feedback to the SysOp }
```

If your word processor has an "overstrike" or "typeover" mode (as opposed to an "insert" mode), enable that option now. What we are going to do is to write the alternative command display over the second copy of each command:

```
=====
M A I N M E N U - please choose an option:
=====
{M - Enter message base }{# - Message base disabled }{F - Enter file base}{# -
File base disabled}
{U - Switch to utility menu }{# - Utility menu unavailable }{S - Switch to SysOp
```

```

menu}{
{0 - Change Your User Options }{# - User options unavailable }{G - Logout - end
this call}{# - Sorry, you can't leave}
{! - Send feedback to the SysOp }{# - No SysOp feedback available }

```

Now, place the special command %switches at the start of each command pair. You will have to switch your word processor back to insert mode:

```

=====
M A I N M E N U - please choose an option:
=====
%M{M - Enter message base }{# - Message base disabled }%f{F - Enter file base}{# -
File base disabled}
%u{U - Switch to utility menu }{# - Utility menu unavailable }%s{S - Switch to
SysOp menu}{
%o{0 - Change Your User Options }{# - User options unavailable }%g{G - Logout -
end this call}{# - Sorry, you can't leave}
%{! - Send feedback to the SysOp }{# - No SysOp feedback available }

```

If you follow this method of creating your custom menu files, you will find it a lot easier to visualize and format them properly. As you can see by this example, creating an alternative menu display is a pretty simple matter. Other codes that you can use when creating custom menu files are:

%POSOFF	— Disable ANSI screen positioning
%POSON	— Restore ANSI screen positioning
%CLROFF	— Disable screen clears
%CLRON	— Restore screen clears
%COLOUROFF	— Disable ANSI colour
%COLOURON	— Restore ANSI colour to user settings
%DOMORE	— Generate a MORE prompt immediately
%SETWIDTH.X	— Set user screen width to x
%PAUSE.X	— Pause for x seconds
%SLOW.X	— Display text with an x tick pause per character
%a0	— colour 0
%a1	— colour 1
%a2	— colour 2
%a3	— colour 3
%a4	— colour 4
%a5	— colour 5
%a6	— colour 6
%a7	— colour 7
%ab	— bold
%af	— flashing (won't show locally)
%ai	— italics
%ao	— ANSI off
%ar	— reverse
%au	— underline

Note: The %a switches are case sensitive.

If you embed actual ANSI sequences into your custom menu displays they will be displayed regardless of the user's account settings, but if you use the ones listed above, they will adhere to the user's account settings and only display if the user has ANSI colour enabled.

In addition to these, you can use any of the user %switches to place personalized user information into your menus. The user %switches were outlined in the chapter on DLG Text Files.

Now let's save this menu file, and create a menu set entry for it. **Save this as an ASCII text file with the pathname: DLGConfig:Menus/MAIN.NOV.1**

You should now log into your DLG system, go to the SysOp Menu, and select the Menu Editor. From the Menu Editor, select the Custom Sets option, and ask to Add a Set (SCCA).

DLG will ask you to supply a number for this menu set (1)

DLG will ask you for a description of this menu set. This description will appear in a list of available menu sets for users to choose from. It should be as descriptive as possible. (Amiga mono menus [English])

DLG will ask you for a language file. When users select this menu set, they are also selecting which language they wish to have DLG use when sending messages and DLG system text. (English)

DLG will ask you for the user level required to use this custom menu set. (1)

Confirm that you want to add this menu set. (Y)

Now let's see this menu in action.

Return to the Main Menu and select the User Options editor. Select item 26, and choose menu set #1.

Once you have done this, return to the Main Menu, and you should see your new menu instead of the default DLG menu. If you don't see the menu right away, type "?" and press RETURN to display the menu.

Congratulations, you have just created and installed your first custom menu. In a similar fashion, you can create alternative menu displays for your message base, file base, utility menu, and so on. Once you have created a menu set in the SysOp Menu Editor, you can add menu files to that set simply by saving them with the correct name, as discussed above. For example, to save a menu file to replace the default file base menu, you would save it as FILE.NOV.1, and it will automatically be considered to be part of the same menu set as the MAIN.NOV.1 file we created here in this tutorial.

Before continuing further, go back to the SysOp Menu Editor and remove this custom menu, or re-edit the custom menu file. It does not have all of the commands that your actual main menu has, and may confuse your users if you leave it in place as is.

Other features of DLG Menu Sets

Once a menu set has been defined, and you have created menu text files to include in it, users can select that menu set as an alternative display. You can extend the concept of alternative text displays beyond just menus using menu sets. Any text file used by the DLG system can also have an alternative determined by the user's choice of menu set. To do this, you append the number of the menu set to the name of the alternative file. For example, if you had an Italian menu set number 6, and you wanted to replace the normal Logout.txt file with an Italian version, you would create the Italian file with the name Logout.txt.6. Any user with the menu set number 6 selected will see the alternative Logout.txt.6 file when logging out of the system.

DLG Configurable Menus

While the custom menu set concept lets you change the look of a menu, the configurable menu concept allows you to change the contents of a menu. With configurable menus, DLG allows you to add or remove commands from any DLG menu on your system.

You change the commands available on a menu with the SysOp Menu Editor. The reference section below describes all of the options available with this editor. We suggest that you refrain from tampering too much with your existing menus until you become more familiar with the DLG system and how things interact. It is fairly easy to completely disable your message or files areas by making

errors when editing your configurable menus. You might even accidentally remove the ability to fix your own mistakes! The thing to remember about configurable menus is to go slowly, changing a few things at a time, and then testing the menu to make sure that everything is functioning properly.

Warnings aside, the process of configuring a menu is not that complex. Each command in a menu consists of a set of parameters which tell DLG how to react to that particular command choice.

These parameters include:

- what module the menu refers to (i.e. "Mess", "File", or "Menu".)
- what letter is used to access the command.
- a brief description of the command.
- whether a command is an executable file, ARexx script, DLG batch file, menu script, command stack, built-in function, or another menu.
- what help file is associated with this menu item.
- whether the menu should remain resident in memory while the command is being executed, or if the menu should exit once the command executable has been activated.
- whether to run the executable in CLI or DLG mode. CLI is an environment similar to a normal CLI window, while DLG mode is similar to a RAW CLI environment.
- whether to pend the display of DLG system messages during the execution of the command and display them just before a prompt.
- Access levels required to access the menu item.
- if the menu item is hidden or not.
- if the command choice requires confirmation from the user before continuing.
- the task priority of the command.
- the activity string for this command.

As you can see, a configurable menu gives you a wide range of options and abilities to customize your DLG environment.

Reference Section:

Here is a listing of the commands available in the SysOp Menu Editor, with a description of each command's function.

Select Menu

This will allow you to select the menu you wish to configure. It will present you with a list of menus and allow you to type a name. DLG's name completion mode will be used at this prompt. To start a completely new menu, type "NEW" at the prompt.

When adding a NEW menu to the system, you will be asked the following questions:

Title for the new menu:

This is the title that should appear on the top of the menu display. If you type something here, you will not only get the title display, but top and bottom dashed lines will be drawn above and below the menu body. This would be suitable for main menus with section titles. If you omit the title, not only will the menu not have a title, but no lines will be drawn above and below the menu body. This is suitable for a shorter 4 column menu like the type you would get in the DLG modules.

Number of columns:

This is the number of columns that the menu display will use. This number represents the number of columns that should appear on a standard 80 column screen. If users screen widths are something other than 80 columns, the number of columns will automatically be adjusted to best suit their display width.

Prompt for menu:

This will allow you to type a custom prompt for the menu. It should be noted that you can use any of the global %switch variables, such as those outlined in the section on text files, and any local %switch variables that are local to the module you are making the menu for.

Menu letter to call when number is typed:

This allows you to specify what letter should be assumed when the user types a numeric value at the prompt rather than a menu entry. For example, Mess is set up so that the '\N' (return) menu entry is called when a number is typed. This calls the Msg_ReadNext routine in Mess with the number the user typed inserted into the command stack.

Module Name:

This is the name of the module that the menu is being designed for. A list of modules will be shown and you will only be able to type in a valid response. For example, if you were designing a menu for the Message base, that would be the 'Mess' module. The module for the main menu of your BBS would be 'Menu'. Each module has a certain set of built-in routines which perform specific functions within that module.

List Menu

Once a menu has been selected, you can list that menu to the screen. The display you see will be identical to how the menu will appear in actual use, except for local %switches.

Add Item

Selecting this option will allow you to add a new menu item to the current menu. The following items of information must be supplied.

Letter for menu entry:

This is the letter that the user will press to activate this menu item. Most characters are supported, and a special character sequence '\n' (note the BACKSLASH) is used to represent the user pressing the RETURN key on the menu. Note that if you try to edit such an entry that you will have to enter '\n' as the menu letter to edit, not 'RET' as it appears in the menu.

Description for entry:

This prompt will allow you to type a description for the menu entry. The recommended length represents what will fit into the display for the current number of columns you have selected for the menu. You can type past the recommended length, but the output will be truncated unless you change the number of columns for the menu.

Command type:

This next prompt gives you several choices as to the type of menu entry:

Executable program

This can be any program that runs in a CLI process and does its input and output to the CLI window. NOTE: programs that use Intuition, the Amiga's graphic environment, or open custom windows will not work as on-line DLG modules. Just type the full name of the executable including the path along with any command line arguments needed to run the program. Any number of global %switch variables and any number of local module %switch variables may be used. These variables will be converted to actual values before the program is executed.

Batch file

This can be any AmigaDOS batch file. Note: A batch file cannot get input from the user. A menu item that uses an AmigaDOS batch file would be one that displays information or executes a utility that does not require user interaction. You can pass user and system information to the batch file using %switches

Sub menu

This will allow you to specify another configurable menu to chain to.

Menu Script

A menu script is very similar to a single menu entry, except it can simulate any number of menu entries in sequence. Menu scripts can contain the %switch variables anywhere in the script. Each entry in the menu script has to be preceded by one of the following keywords:

- EXE <path/filename> <args> - an executable program
- BATCH <path/filename> <args> - a batch File
- SUBMENU <menuname> - a sub menu
- MENUSCRIPT <path/scriptname> - another menu script
- STACK <command stack> - a command stack
- AREXX <path/filename> <args> - an ARExx script
- BUILTIN <command name> - a built in command

The following commands are modifiers and will affect all commands following in the script. They basically represent all of the settings that can be set for a normal menu.

- ACTION <action string> Activity string for user
- ULEV <upper user level limit to use command>
- LLEV <lower user level limit to use command>
- PRI <priority>
- CHAIN - the next program will chain and the rest of the menu script will be ignored.
- OVERLAY - following commands will overlay
- BCPEND - broadcast messages will pend until a prompt
- BCRESUME - Enable 5 line broadcast messages
- CLI - Will set CLI mode for following commands
- NOCLI - Will set DLG mode for following modules
- PAUSE - Will cause DLG to present a [Press Return] prompt after each of the following commands

Command stack

A command stack is a sequence of characters that simulate user input. This can be used to automate a complex sequence of menu selections for the user.

ARExx script

This will allow you to execute an ARExx script. Note that you must have ARExx installed on your system to be able to use this feature. ARExx is a part of Workbench 2. To configure a menu entry to call an AREXX script, simply select the AREXX item type, and enter the full path to the AREXX script. DLG will default to looking for a script in the REXX: directory with the extension ".drx", but in order to properly edit the script from within the DLG menu configuration software you should specify the complete path, filename, and extension.

Your ARExx scripts can do anything they would normally do when run from the CLI. You can use the regular ARExx commands for getting input from the user and displaying output. You can also use the ARExx "Address command" to execute CLI programs. In addition, DLG gives you the ability to call the built-in functions of the module that is currently loaded. The syntax for doing this is:

<built-in> "<stack>"

where <built-in> is the name of the built-in function, and <stack> is a command stack to be inserted into the user's command stack *before* the built-in function is executed. For example, if an ARexx script was executing and the user was in their private directory, the ARexx command:

```
Msg_Write "l;sysop;FeedBack"
```

would load up the message editor ready to edit a local message to Sysop with a subject of FeedBack.

DLG also provides several other ARexx commands:

CLI

Put DLG in CLI mode (RAW off, ECHO on - used for CLI programs and most internal ARexx operations).

NOCLI

Put DLG in NoCLI mode (RAW on, ECHO off - used for DLG modules).

OVERLAY "<program>"

OVERLAY the specified <program> (just like the OVERLAY option in the menu configuration).

CHAIN "<program>"

CHAIN to the given <program> (just like the CHAIN option in the menu configuration). Note that you can use CHAIN to move from one DLG module to another while still staying in the same ARexx script.

TRANSLATE "<string>"

Translate the %switches in <string> using both global and local (to the module) switches. The result of the translation is returned in the ARexx variable "RESULT" (note that you must specify "OPTIONS RESULTS" in your ARexx script for this to work).

NOTE: It is not currently possible to change DLG's variables from within an AREXX script. This ability is planned for a future version of DLG.

Built-in function

This will allow you to call any of the built-in functions in any of the modules. The names of the built in functions will be displayed in a list and you will only be allowed to type a valid choice. Built-in functions provide functions such as file downloading, message reading, message editing, etc.

The following is a list of the built-in functions from Mess and File:

MESS:

- | | |
|----------------|---|
| DisplayMenu | - This function is available in all DLG menus. It displays the current menu. |
| Help | - This function is available in all DLG menus. It displays the current menu and prompts the user to choose a command letter. The help text associated with that menu item is then displayed. |
| MSG_ChangeArea | - This function will allow a user to select a new message area. If selected without an accompanying area number (as in "A20") then the command will prompt the user to enter an area number, or optionally display a list of all available message areas, and then prompt again for an area number. |
| MSG_ContRead | - This function will enable the "continuous read" mode of the message base. The user will be prompted to provide settings for ANSI stripping, and more prompts. |

MSG_Correct	- This function will enable users to edit messages they have just read.
MSG_DeleteAll	- This function will enable users to delete all mail from their private message areas. This function will only be active if users are IN their private message area at the time it is invoked.
MSG_EditSig	- This function enables users to edit their signature files. Users will be presented with a list of possible signatures to edit - local, UseNet, EchoMail.
MSG_Forward	- This function will allow a user to move or copy the message just read to a different message area, or to another user.
MSG_FwdRead	- This function allows the user to select the forward reading mode (the default).
MSG_Kill	- This function allows the user to delete the message just read.
MSG_LexCheck	- This function allows the user to "lex check" the message just read.
MSG_ListReaders	- This function lists the names of the users who are regular readers of the current message area.
MSG_NewScan	- This function will scan the message areas listed in the user's global scan preferences, looking for new messages.
MSG_PvtArea	- This command will switch a user to their private message area.
MSG_ReadNext	- This function will allow the user to read the next message in the area. If there are no more messages in the current area, this command will take the user to the next available message area from their global message area list. The "next message" is dependent upon the reading direction, and reading mode chosen (tag , thread, or normal).
MSG_ReadOrig	- This function will allow the user to read a message that the message just read is a reply to.
MSG_ReadReply	- This function will allow the user to read a message that is a reply to the message just read.
MSG_ReadTagged	- This function will allow the user to read messages that they have tagged for future reading, or have been tagged for them by the BBS when they were entered into the system.
MSG_Reply	- This function will allow the user to reply to a message that they have just read.
MSG_ReRead	- This function will allow the user to re-read a message that they have just read.
MSG_RevRead	- This function will allow the user to read messages in reverse order.
MSG_Search	- This function will activate the header search feature. The user will be prompted for a search string if one was not provided with the command. If the search string is empty then the search feature is disabled.

MSG_SelectSig	- This function will allow a user to select a different SIG. If no accompanying SIG number is specified (as in "S1") then the command will prompt the user to enter a SIG number, or optionally display a list of all available SIGS, and then prompt again for a SIG number.
MSG_SkipThread	- When in thread read mode this command will allow the user to skip threads they are not interested in. All messages remaining in the current thread will be skipped by the message reader.
MSG_TagRead	- This command activates a header scan mode where only the headers of messages are displayed. Readers can then tag messages that they want to read in full in tag read mode.
MSG_ToggleThread	- This function toggles thread reading mode on and off. In thread reading mode, all messages that have the same subject line are considered to be part of a "thread", and will display one after another. In normal reading mode, messages are read in the same order that they arrived in the message area. Topics are interwoven amongst each other, and conversations can be more difficult to follow in normal reading mode. When in thread reading mode you have the option of skipping message threads you are not interested in reading.
MSG_UpdatePtr	- This function will update the reader's high message pointer to the highest message in the area. This provides new users with a quick way of "catching up" to the most current messages on your system.
MSG_Write	- This function enables users to compose messages, with the use of the editor of their choice (determined in their user options).
MSG_WriteBltn	- This function enables users to compose bulletins, with the use of the editor of their choice. Bulletins differ from messages in that they appear automatically to all users as they log into the system, and that they have an "expiry date", after which they are removed from the system.
FILE:	
DisplayMenu	- This function is available in all DLG menus. It displays the current menu.
Help	- This function is available in all DLG menus. It displays the current menu and prompts the user to choose a command letter. The help text associated with that menu item is then displayed.
File_ChangeArea	- This function will allow a user to select a new file area. If selected without an accompanying area number (as in "A20") then the command will prompt the user to enter an area number, or optionally display a list of all available file areas, and then prompt again for an area number.
File_Comment	- This function will allow a user to add a comment to the file description just read, using the editor they have chosen in their user options.

File_Download	- This function will allow a user to download a file. If the user has just read a file description, the option will be to either download that file or files that the user has tagged for batch downloading. If the user has not just read a file description, then the system will assume they want to download the tagged files, or if there are no tagged files, prompt the user for the name or number of a file to download from the current file area. If the user has a preset download file transfer protocol, then that will be used automatically. If the user has no preset protocol, then DLG will prompt the user to select one from a list of all protocols available.
File_Edit	- This function will allow the user to edit the file just read. Edited items include the file description, file name, and file size.
File_EditSig	- This function enables the user to edit their signature files.
File_GlobalList	- This function lists all of the files in all of the file areas that the user has selected in their user options. The user will be asked for listing options - forward, reverse, alphabetical forward, alphabetical reverse, all files, new files, since last call, since # of days, since date, in a given range of dates, search for a specific filename, or by filename and description.
File_Kill	- This function will allow a user to delete the file and description just read.
File_List	- This function is very similar to File_GlobalList, except that it functions only in the current file area, not all files that the user has selected to scan.
File_ListBatch	- This function allows the user to list all of the files that they have tagged for batch downloading.
File_NewScan	- This function will search all of the file areas that the user has selected in their global file area list in the user options, for new files. A new file is one that the user has not "seen" yet, not new files since last log-in.
File_PvtArea	- This function will take the user to their private file area.
File_Read	- This function will enable the user to read each new file description in an area. On the default DLG system, this is the command implied when the user presses RETURN at any prompt. If a number is supplied, then the file with that number is read instead of the next available file.
File_RemoveBatch	- This function will enable the user to delete some or all of the files that they have tagged for batch download.
File_SelectSig	- This function will allow a user to select a different SIG. If no accompanying SIG number is specified (as in "S1") then the command will prompt the user to enter a SIG number, or optionally display a list of all available SIGs, and then prompt again for a SIG number.
File_Tag	- This function adds the file just read to the user's list of files for batch download. If the user has not read a file description, they will be prompted for the name or number of the file to tag.

File_Transfer	- This function enables the user to transfer the file just read to another file area or to a user's private file area.
File_Upload	- This function enables the user to upload a file to the current file area, or to a user's private file area. If the user has an upload protocol preference selected in their user options, it will be used automatically. If not, the user will be prompted to select a file transfer protocol from a list of all available protocols.
File_Validate	- This function enables the user to validate an uploaded file. The validated file is moved from the designated validation file area to the area it was originally uploaded to, or to a file area of the user's choice.
File_ViewArchive	- This function enables the user to list the files contained within the current file, if it is one of the configured archive file types.
File_ViewNext	- This function enables the user to see the description of the next file in the current file area, or to go to the next available file area from their list of global file areas.

Help file:

This prompt allows you to set the file that will be displayed when the user asks for help on the menu entry. In most cases you will choose the default that DLG offers.

Chain or Overlay:

This prompt is very important. Some of the larger DLG modules are designed to be CHAINED from a menu. If you select chain, the menu will call the executable program and then exit. It will be the responsibility of the module to chain itself back to the menu when it exits.

If you respond NO to the CHAIN option, the command will overlay when it is loaded. Overlaying a module will use a little more memory since it will cause the menu to stay loaded while the command is being executed. In this mode the command must only perform a simple EXIT when it has completed. Once the command exits, the menu resumes from exactly where it left off.

If you have set a DLG program to overlay instead of chain, and you notice that you have to exit a menu twice, you will have to change the menu setup of the command to chain instead. This happens because a command designed to chain will reload the menu automatically when it has been exited. If such a command has been set to overlay, you will have two copies of the menu, and you will end up having to exit both of them.

ALL non-DLG CLI programs will have to be run in overlay mode and most large DLG modules have to be run in chained mode. Please see the reference section if in doubt.

Run in CLI mode:

This prompt allows you to set the low level I/O environment for the command. CLI mode is an environment very similar to a normal CLI window and most CLI applications will be run in this mode. The alternative to this mode is DLG mode. This is very similar to a RAW CLI environment and allows for things such as hot key input. All DLG modules will run in DLG mode and not CLI mode.

Confirm prompt:

If you answer YES to this prompt, DLG will ask the user if they are sure before it goes ahead and executes the command. For example, the Goodbye command on most menus has this option enabled.

Pend broadcast messages:

Whenever a user logs onto the system, or when a message addressed to a user is placed on the system while that user is on-line, DLG will generate a system message announcing this fact. Also, messages can be generated by the broadcast module, or by the conferencing software when a new room has been created by a user. Answering YES to this prompt will make all DLG system messages wait until the user is at a prompt, and display on a single line before the prompt. This is true only if the module running at the time is a DLG module. If you answer NO, then messages will display at any time in the standard 5 line format.

Hidden menu item:

You may wish to have HIDDEN menu entries that really exist and accessible by the user but will not display in the menu.

Low access level:

This is the lowest access level that this command will show up and be allowed for.

High access level:

This is the highest access level that this command will show up and be allowed for.

Custom log value:

If you have any custom log entries defined via the GenConfig module, you can have DLG automatically insert them into the log each time this command is called.

Activity string:

This is the activity string that will be written to the t:<portname>.user file. This activity string is what will show up for a user when another user selects the WhoIsOnline menu option. You can use the %switches in this activity string.

Priority:

This sets the current priority for the menu entry. If you press RETURN at this prompt, you will get a default priority mode, which can change from port to port. If you use a priority other than the default priority, then the following rules apply:

If a priority is used on an overlayed module, a built-in command, or a batch file, the priority of the task will be temporarily changed to the new priority while the command is being run, and set back to default afterwards.

If a command is a change of menus or a chained module, then the new priority will become the default priority for the rest of the session.

Edit help file:

This allows you to edit/create the help file for this menu entry.

Local variables:

Any command strings, activity strings or MenuScripts can use the standard global %switches, as well as local %switch variables to the module that the menu is created for.

The following is a list of local %switch variables for Mess and File.

Mess:

- %Msg_MsgNumber - The number of the current message
- %Msg_AreaNumber - The number of the current message area

%Msg_AreaName	- The name of the current message area
%Msg_Direction	- The user's current reading direction ("Forward" or "Reverse")
%Msg_ThreadMode	- The user's current thread-reading status ("On" or "Off")
%Msg_TagRead	- The user's current tag-read status ("On" or "Off")
%Msg_LowMsg	- The number of the low message in the current message area
%Msg_HighMsg	- The number of the high message in the current message area
%Msg_SigNumber	- The number of the user's current SIG ("0" if no SIG selected)
%Msg_SigName	- The name of the user's current SIG ("NO SIG" if no SIG selected)
%Msg_UserHighMsg	- The user's high message pointer in the current message area
%Msg_AreaType	- The type of the current message area ("LOCAL", "NETMAIL", "ECHOMAIL", "USENET", "PRIVATE")
%Msg_Zone	- The current port's FidoNet ZONE number
%Msg_Net	- The current port's FidoNet NET number
%Msg_Node	- The current port's FidoNet NODE number
%Msg_Point	- The current port's FidoNet POINT number
%Msg_From	- The name of the user the current message is FROM
%Msg_UFrom	- The underscored name of the user the current message is FROM
%Msg_To	- The name of the user the current message is TO
%Msg_UTo	- The underscored name of the user the current message is TO
File:	
%File_FileNumber	- The number of the current file
%File_AreaNumber	- The number of the current file area
%File_AreaName	- The name of the current file area
%File_LowFile	- The number of the low file in the current file area
%File_HighFile	- The number of the high file in the current file area.
%File_SigNumber	- The number of the user's current SIG ("0" if no SIG selected)
%File_SigName	- The name of the user's current SIG ("NO SIG" if no SIG selected)
%File_UserHighFile	- The user's high file pointer in the current file area
%File_FileName	- The name of the current file
%File_FilePath	- The full path to the current file
%File_From	- The name of the person who uploaded the current file
%File_UFrom	- The underscored name of the person who uploaded the current file.

DLG SysOp Editors

This chapter outlines the purpose and function of each of the various editors available in the SysOp menu. In previous chapters, you have had experience in using several of the most important editors. In this chapter, we are going to have an overview of each of the editors, but the detailed functioning of each of the editors can be found in the chapters that pertain to the topic of each editor. For example, the chapter about DLG's message areas contains tutorials and explanations of the Message Base Editor, and the User Message Area Access Editor. The information there will not be repeated here, except in a general overview form. Other editors have not been covered in detail in previous chapters, so this chapter will contain more detailed information on them.

The Main SysOp Menu

Here is a listing of all of the various editors that you will find in DLG's main SysOp menu:

```
Select => $
Sysop Sub-System
-----
[D] Define / Edit Message Areas      [F] File Area Define / Edit
[E] Edit Message Area Access        [R] Revise File Area Access
[U] User Account Edit               [A] Group Account Edit
[P] Port Configure                  [N] Network Mail Configure
[C] Configure Menus                 [T] Transfer Area Maintenance
[G] General Configuration           [S] SIG creation/edit
[M] Main Menu                       [?] Display menu
[H] Help
-----
Select => █
```

Define / Edit Message Areas

This editor, which is featured in tutorials in the chapter on DLG Message Areas, is what you will use to create or modify message areas on your system. When you create a message area, you will be led through a list of prompts that will help you define the attributes of the area. This editor will also allow you to modify the attributes of an area once it is created, or to delete the area altogether. Attributes of an area can include: entry user levels, activity user levels (such as message writing, re-editing, forwarding, etc.), message area capacity, renumber trigger, special or auto access, and so on. All of these topics are covered in detail in the chapter on DLG Message Areas.

File Area Define / Edit

This editor is very similar in function to the Message Area Editor, except that its purpose is to facilitate the creation, modification, or deletion of file areas. When you create a file area, you will be led through a list of prompts that will help you define the attributes of the area. This editor will also allow you to modify the attributes of an area once it is created, or to delete the area altogether. Attributes of an area can include: entry user levels, activity user levels (such as uploading, downloading, transferring, and so on), file area location, special or auto access, and so on. All of these topics are covered in detail in the chapter on DLG File Areas, where this editor is featured in tutorial sessions.

Edit Message Area Access

This editor will allow you to edit user access to your message areas. Normally, when you create a message area, users will gain access to the area and the features in that area based on their user levels. To modify the access that any particular user has to the message areas in general, you would

modify their user level in the User Account Editor. There will be times when you need to make adjustments to user level access that cannot be addressed by user level alone. That is where the Message Area Access editor comes into play. With the Message Area Access editor, you can add or remove users from auto-access areas, and special access areas. You can also copy the user-list from one message area to another, making the creation of several areas with similar membership lists easy. The Message Area Access Editor will also allow you to list the areas that are available to a particular user, or list the users in a particular area. The Message Area Access Editor will also allow you to adjust individual user access within a given message area, by adding or removing various access features from the area's default access levels. For example, you could set an area up as read only, and then add the write access level only to those users for whom you want to have the ability to write messages.

Revise File Area Access

This editor is very similar in function to the Message Area Access Editor, except that it performs its work on file areas. You can add or remove users from the user list of a given file area, copy the membership list from one area to another, and edit the user access to features within a given file area.

User Account Edit

DLG's User Account Editor allows you to modify many of the attributes that make up a user account. From the User Account editor, you can create a new user, delete an existing user, modify an existing user's account, or validate new users with the use of prepared template files. The User Account Editor also provides the facilities for the creation and maintenance of the user validation templates that you will use when validating users. Let's examine each of the User Account Edit commands in detail:

List users

This command will simply list all of the users on your system, in a nice three column format.

Edit user

When you select this command you will be prompted for the name of a user to edit. DLG's smart completion routine will guess ahead as you type the characters of the user's name, helping you zero in on the user you wish to edit in the fewest possible keystrokes. When you press ENTER, you will see a screen full of information

```

TL0: JAMES HASTINGS-T B:19200 ON: 0:30 IA:1425 T:Aniga
[1] Name - JAMES HASTINGS-TREW [1] Alias -
[2] Address - [3] Terminal - Aniga
[4] City - [5] Password -
[6] State/Prv - [7] Joined - Fri 9 Jul 92 21:06
[8] Country - [8] Lst Login - Sat 10 Jul 92 0:14
[9] Pst Cd - [9] BirthDay - 0/0/0
[10] Phone No. -

[11] Help Level - Novice [12] User Level - 255 [13] Screen Len - 23
[14] More Prmpt - No [15] Dly Limit - 1440 [16] Scrn Width - 80
[17] Ansi Flag - ---- [18] Sesn Limit - 1440 [19] Msgs Read - 0
[20] Hot Keys - No [21] Used Today - 14 [22] Files Upld - 3
[23] Lex Flag - No [24] Num Calls - 6 [25] 'K' Upld - 220
[26] Pop Screen - No [27] Dir Limit - 10000 [28] Files Dnld - 2
[29] UUCP Priv - NONE [30] Msgs Wrtn - 1 [31] 'K' Dnld - 37
[32] Tm Online - 1 [33] Lst MArea - 999 [34] Sysop Pgs - 0
[35] Bltn Write - Yes [36] Lst FArea - 99 [37] U/D Ratio - NONE
[38] Credit - 10000 [39] Protocol - D:- U:- [40] Capture - NONE
[41] Editor - 4 [42] NetMail - WS [43] Archiver - 0
[44] Menu Set - 0 [45] Char Set - 0
[46] Login stack - [47] Apply template [48] User notes

Number to edit [RETURN to exit] => █

```

Each item of information that you can edit will have a number beside it. Note that there are a few pieces of information that you will not be able to edit. Some information is permanent - the user's name and birthdate, for example. Some information is editable by the user, and so it is not available here for editing at all. To select an item of information to edit, simply type the number corresponding to it. Item 47 will allow you to apply a TEMPLATE of editing options to an account. A template is a special user account entity that will allow you to edit many items at once.

One item of interest is the notes option. As you edit a user's account, you can enter some notes about that user for future reference. These notes will go to special directory called DLGConfig:Notes. These notes are displayed when you enter chat mode with the user. You can also call up note information at any time during chat mode with a user by pressing CTRL-N. This is a handy way of reminding yourself who the user is, and what business you may have with them.

Add user

The add user command will allow you to create a new user account. Most new users will create their own accounts the first time they log into your system. However, should you need to create an account for a friend or for test purposes, you can create the account here. The prompts and questions asked are very similar to the new user application that is filled out by your users on their first visit to your system.

Delete user

This command will prompt you for the name of a user to delete. The smart completion mode will guess ahead for you, so that the user name can be completed in the fewest possible keystrokes. The command will ask you if you are sure you want to go ahead with the deletion. If you have a large system, user deletion can take a considerable amount of time, as DLG will look through all of the message and file area membership lists and delete the user from each of them. Deleting a user will also remove that user from any groups that have been defined.

Purge users

This command will allow you to delete old or deadwood accounts. You will be prompted to provide the number of days back to look for old, inactive accounts. Any user who has logged into your system within that number of days will be safe from the purge, but any user who has not logged into the system in that time will be removed. You will also be prompted for a maximum user level to purge. Users whose levels meet or exceed the threshold you set here will not be purged, no matter

how inactive their accounts are. You will be asked if you wish to be prompted on each deletion, or have the deletion proceed automatically. We suggest that the first time you do a purge that you allow DLG to prompt you on each deletion. This way you can get a feel for how old the accounts are you are deleting, and which users will be affected by the purge. When an account meets all of the given purge criteria, and you have prompting turned on, DLG will list that user's account information on the screen, and ask you if you wish to delete that user. The purge can be cancelled at any time with a CTRL-C. Users names will not be deleted from message or file areas, or groups, until the end of the purge session.

Validate user

When a new user logs into your system, they are given the attributes of the NEW USER template supplied with your DLG system. New users are also flagged as "unvalidated" user by DLG. When you log into your DLG system, you will be reminded that unvalidated users exist on your system. The process of validating a user consists of applying a different template to that user's account. As you validate each user, their user account information is listed on your screen. You will have the opportunity too see their voice telephone number. It is common practice to call prospective users by voice to "validate" the information in their user account - i.e. make sure people are who they say they are. Trouble-makers may often try to disrupt the smooth operation of your system by logging in under false names and user information. As you see the user information on your screen, you will have the opportunity to either edit the user account information, apply an existing user account template, delete the account, or ignore it and leave it unvalidated.

Batch edit

This command will allow you to edit several users at once, by applying an existing user account template, and provides a number of ways of selecting users to edit. You can also elect to apply only those items in the template that would upgrade the user accounts, or to apply the entire template, regardless of setting. You can choose to batch edit all users on your system, edit users listed in a standard text file that consists of names of users (one per line) all the users in a particular group, all users of a given user level, or you may choose to enter user names one at a time, using the smart completion mode.

Add template

This command will allow you to create a new user account template, which is used in many of the user account editing operations. Here is a list of the pieces of information that you will be prompted for:

Name of template (this is the name that will appear in the list of templates. You should have a template defined for you in the default DLG installation called "NewUser". DLG uses this template when creating new user accounts.)

User level (this is the user level to apply)

Upload/Download ratio (this is the user level to enforce)

Auto screen pop (whether or not you want DLG to automatically open a screen when the user logs in)

Enable bulletin write (allow the user access to write bulletins)

Daily call limit (the number of minutes that the user can use the system in one day (0 - 1440))

Per call limit (the number of minutes that the user can use the system in any one call (0 - 1440))

'K' uploaded (the number of kilobytes uploaded by the user - useful for working with particular kinds of file upload/download ratio situations)

'K' downloaded (the number of kilobytes downloaded by the user - useful for working with particular kinds of file upload/download ratio situations)

UUCP Access: [C] Client [W] Write Access [N] None (If an account is a UUCP client, then that account will trigger a UUCICO session once the name and a password has been given. A UUCP

client account does not have access to the rest of the DLG system. This type of account is used to set up automate transfers of Mail to and from UseNet systems. UUCP Write access allows a user to write UseNet mail from their private mail area.)

User directory size limit in 'K' (the amount in kilobytes of disk space available for storage in the user's private account directory)

Credit units (the number of NetMail credit units the user's account will have)

NetMail write access (indicates that a user can write NetMail to users on remote systems)

NetMail SYSOP access (indicates that a user can write NetMail to users on remote systems, and include file attaches, requests, or crashmail options)

Del template

This command will allow you to delete a template that you have created.

Edit template

This command will allow you to edit an existing user account template

Group Account Edit

Group editor			
[L] List groups	[A] Add Group	[D] Delete group	[U] List users
[N] Add user	[K] Delete user	[M] Exit	[?] Display menu
[H] Help			
Select => █			

Group accounts allow you to treat lists or groups of users as single entities. You can send a message to a group account, and each member in that group will receive that message. You can also edit user accounts by group in the User Account Editor. This editor will allow you to list all existing groups, add a new group, delete an existing group, list the users in a group, add a user to a group, or delete a user from a group. There is a tutorial on the Group Account Editor in the mini-tour at the beginning of this manual.

Port Configure

Select => P			
Port Configuration - Main Menu			
[P] Port submenu	[D] Displays	[G] Global sets	[O] Modems
[C] Add computers	[L] List computers	[M] Exit	[H] Help
[?] Display menu			
Select => █			

The port configuration editor is where you configure and modify many aspects of the DLG system. There are a number of sub-menus to this menu, where you can add, modify, or delete information about your modems, display preferences, port configurations, global preference settings, and so on. There is a full tutorial on the Port Configuration Editor in the chapter on setting up your DLG System.

Network Mail Configure

FidoNet configuration

[E] Edit config [M] Exit [?] Display menu [H] Help

Select => █

This simple menu will allow you to edit your NetMail configurations. Here you will be able to enter the following information:

Zone number
Net number
Node number
Point
Origin line
Nodelist path
Default NetMail flags: Crash mail Kill send

The default NetMail flags define what the defaults for sending a NetMail message will be. You can over-ride these defaults at the time a NetMail message is being composed. DLG can run as a point. If you specify a point number in your default FidoNet configuration, DLG will automatically append a FMPT kludge line to all outgoing NetMail messages sent from the Mess or Forward modules.

Configure Menus

Menu editor - Current menu [NONE]

[S] Select Menu [L] List Menu [A] Add Item [D] Delete Item
[E] Edit Item [O] Change Order [F] Format Menu [R] Remove Menu
[U] Undo changes [C] Custom Sets [M] Exit

Menu editor command => █

This is the menu configuration editor. There is a full tutorial on the menu configuration in the chapter on Configuring your Menus.

File Area Maintenance

File editor

[B] Batch upload [T] Turbo upload [A] Select area [E] Edit file
[F] Freshen [G] Global Edit [L] List files [M] Exit
[H] Help [?] Display menu

Select => █

This is the file area maintenance editor. There is a full tutorial on this editor in the chapter on File Areas

General Configuration

General configuration editor			
[L] Logs	[E] Editors	[P] Protocols	[A] Archivers
[C] Char Sets	[F] File Paths	[M] Prev Menu	[?] List Commands
General configuration editor => ■			

The General Configuration Editor allows you to configure many miscellaneous aspects of your DLG system. It is here that you can configure custom log entries, add new text editors, add file transfer protocols, add new file archivers, add new character sets, and add to the global file paths for your file areas.

Log Configuration

A log is a record of the activities that take place on your board. Each activity has a code which is used to record that activity in the log. Codes 0 to 127 are reserved for use by DLG. Codes 128 to 255 can be defined by you using this editor to represent custom log entries. Log entries can be used in one of two ways. You can assign a log code to a particular menu entry, and each time that menu entry is used, that code will be recorded to the log. You can have a batch file use the WriteLog program to record custom log entries for events that are controlled by batch files or TPTCron. Here is a listing of the command available for working with log levels:

Set Levels

This command allows you to set user-levels to the various activity codes. When the logs are displayed, activity codes that have a higher user level than that of the user viewing the logs are hidden. This command will only allow you to edit the levels of the hard-coded commands. To edit the level of a custom log code, use the Edit Custom command instead.

List Custom

This command will list all of the custom-defined log codes, with the user levels associated with each log entry code.

Add Custom

This command will allow you to create a new custom log entry code. You will be prompted for the number for this log code, a description for the entry that will appear in the log listings, and a user level needed to see this entry.

Delete Custom

This command will allow you to delete a custom log entry that you have previously defined. You will be prompted for the log entry code to delete, and asked to confirm the choice.

Edit Custom

This command will prompt you for the number of a custom log code, and list the defined description and user level. You will be prompted to change these attributes.

Editor Configuration

This is where you can add or redefine text editor for use in various parts of your DLG system. There are a number of commands available here to facilitate the maintenance of your DLG editors. We suggest that you examine the existing editor configurations to gain an understanding of how to add new ones or modify the existing ones:

List Editors

This will list the number and description of each of your defined DLG text editors

Add Editor

This command will allow you to define a new editor for use on your system. You will be prompted to provide a number for the editor, and a description of the editor. Next you will be asked for the arguments to call the editor. The three special switches ~b ~h ~r are for the bodyfile, headerfile, and replyfile respectively. All of the other DLG switches (as supported by DF) can also be used here. This means that you could pass such things as the screen width and height to the editor.

Default settings for the dialog editors are as follows:

```
LineEditor: DLG:LineEdit ~b 10000
ScreenEdit: DLG:ScreenEdit ~b ~h ~r %SCWIDTH %SLENGTH ibmfont %ANSI
```

The next thing you will be asked for is the handler flag settings for the editor. DLG comes with its editors configured properly already. If you are adding a third party editor to DLG, it will come with appropriate handler flag settings. These flags are described in the Reference chapter under the TFlags command.

The final thing you will be asked for is a user level required to select the editor. Here is an example setup for the provided SysOp text editor:

```
Num  Editor
-----
[ 1] Line Editor (Standard ASCII Based)
[ 2] Full Screen Editor (ANSI - IBM Characters)
[ 3] Full Screen Editor (ANSI Amiga Font)
[ 4] Sysop Editor
-----

Enter number to edit => 4
Data for editor number [4]
[1] Name: Sysop Editor
[2] Call string: DLG:ScreenEdit ~b ~h ~r %SCWIDTH %SLENGTH %ANSI sysop
[3] Flags
[4] User level required: 255

Select => █
```

If you are adding a third party editor to your DLG system, use the suggested command line entry suggested for that editor. For a discussion of the command line options for DLG's line and screen editors, see the chapter on DLG Executables.

Edit Editor

This command will allow you to change the setup for an editor that you have on your system.

Delete Editor

This command will allow you to remove an editor from your setup.

Protocol Configuration

This editor will allow you to configure, add, or remove file transfer protocols from your system. We suggest that you examine the existing protocol definitions for examples of how to add new protocols to your system. Here are the available commands:

List Proto

This will list all of the protocols on your system, with their capability flags and transmit/receive efficiency ratings.

Add Proto

This will allow you to add a new file transfer protocol to your system. You will be prompted to supply the following information:

Name of the protocol

Flags - these indicate the capabilities of the protocol (D for download, U for upload, B for batch transfers, R for resume transfers, F for filename required) Some protocols will have some or all of the available capability flags, and it is important to list these properly. For example, Zmodem can download, upload, batch transfer, resume transfer, but does not require the user to enter the filename. Therefore, the flags for Zmodem would be DUBR. DLG will ask you the flag questions one at a time.

Command lines for single file send, single file receive, and batch send (if applicable) - these depend upon the particular transfer protocol software that you want to configure. DLG comes pre-configured with Xmodem, Xmodem 1K, and Zmodem, and may come with others. If you are configuring a third-party file transfer protocol, then you will need to refer to the documentation that comes with that software. If you are configuring a protocol that uses the XPR.library, you will need to use XPRTransfer (as outlined in the reference section) in accordance with the documentation that comes with that new protocol.

Send and Receive efficiency percentage - these are used by DLG as part of the calculation for how long a file will take to download. They displayed in protocol lists to indicate relative transfer speeds.

Send and Receive success codes - These are AmigaDOS return codes that the file transfer protocol software will send back at the end of a successful transfer. Currently, all of the DLG file transfer protocols return a 0 for a successful transfer.

User level required - This is the user-level required by any user wishing to use this file transfer protocol.

Data for protocol [Z]

```
[ 1] Name:          Zmodem
[ 2] Flags:         DUBR-
[ 3] Single send:   DLG:DLGZmodem -sf ~f
[ 4] Single receive: DLG:DLGZmodem -r
[ 5] Batch send:    DLG:DLGZmodem -sbf,~b
[ 6] Send efficiency: 96 percent
[ 7] Receive efficiency: 96 percent
[ 8] Send success code: 0
[ 9] Receive success code: 0
[10] User level required: 1
```

Select => █

Edit Proto

This command will allow you to modify the setup for an existing protocol.

Delete Proto

This command will allow you to remove the setup for a protocol.

Archiver configuration

This editor will allow you to add or modify the configuration of the various file compression programs used by such DLG features as mail packing, and archive viewing from the file areas. We suggest that you examine the existing archiver definitions to gain an understanding of how to create new archive definitions or to add new ones to your system. Here is a list of the available commands:

List Arc

This will list all of the available archivers on your system, with an estimate of their relative file compression efficiency.

Add Arc

This command will allow you to add a new file archiver to your system. New file archivers are created at a rate of about one a year, each one claiming to be the latest state of the art in file compression. This feature of DLG will allow you to stay current with the latest developments in file archiving technology. You will be prompted for the following information:

Name - this is the name that DLG will use to show this archiver in lists that the user can choose from.

Compress String - this is the command line required to access the compress function of the particular archiver software. This is passed two parameters by DLG - ~d which stands for the name of the destination archive, and ~s which stands for the list of files to archive.

Success code - this is the AmigaDOS return code sent by the archiver software at the successful completion of the compression.

Decompress String - this is the command line required to access the decompress function of the particular archiver software. This is passed one parameter by DLG - ~d which stands for the name of the archive to be decompressed.

Success code - this is the AmigaDOS return code sent by the archiver software at the successful completion of the decompression.

View String - this is the command line required to access the view function of the particular archiver software. This is passed one parameter by DLG - ~d which stands for the name of the archive to be viewed.

Success code - this is the AmigaDOS return code sent by the archiver software at the successful completion of the archive viewing.

Integrity Check String - this is the command line required to access the archive integrity checking function of the particular archiver software. This is passed one parameter by DLG - ~d which stands for the name of the archive to be checked.

Success code - this is the AmigaDOS return code sent by the archiver software at the successful completion of the checking.

Filename extension - this is the ".extension" that is normally used by this particular archiver. DLG uses this extension to figure out what archiver to use on the file. If the archiving software can create more than one kind of extension, you will need to configure a new archiver entry for each of the extensions possible. For example, LHA can produce either .LZH or .LHA archives or view .RUN archives.

```

Data for archiver number [3]
[1] Name: Lharc
[2] Compress string: lharc a ~d ~s
    Success: 0
[3] Decompress string: lharc e ~d
    Success: 0
[4] View string: lharc v ~d
    Success: 0
[5] Integrity check string: lharc t ~d
    Success: 0
[6] Filename extension: .lzh
[7] Average compression: 85
[8] User level required: 1

Select => █

```

Average compression - this is for informational purposes only, and represents a best guess.

User Level required - This is the user-level that a user has to have to be able to use this archiver.

Edit Arc

This command will allow you to modify the configuration for an existing archiver.

Delete Arc

This command will allow you to delete the configuration for the given archiver.

Character Set Configuration

DLG supports multiple character set definitions. By default, DLG uses the ISO Latin 1 character set native to North American Amigas. European and other latin alphabet languages have national characters or accented characters which are mapped differently to the ISO Latin 1 character set. A character set file consists of 512 bytes - 256 bytes that represent the translation from the national character set to ISO Latin 1, and 256 bytes that represent the translation from ISO Latin 1 to the national character set. All messages are stored on a DLG system in the ISO Latin 1 character set, except when specified otherwise, and are translated by the chosen character set file only when written and read. Users can individually choose any character set that you have on your system. The Character Set Configuration editor will allow you to list, add, or delete character sets from your system. You can create your own character set translation file by editing or creating a new character set translation matrix using this editor. **Note: this is a sensitive procedure, so do not attempt to create a new character set file unless you know exactly what you are doing.** As new character set files are created they will be made available on the DLG support BBS's systems around the world.

File Path Configuration

This topic and editor were covered at length in the chapter on File Areas. Briefly, file paths are global path assignments on which files for any file area can be found if they are not present in the file areas themselves. You can list, add, or delete your global file assignments as your storage capacity and needs change. TPTFreq will also support these global file paths. For CD-ROM owners, this means that the contents of their CD-Rom can be placed on-line for file request.

SIG creation/edit

The SIG configuration editor is where you can define Special Interest Groups for your message or file areas. A SIG consists of a list of message (or file) areas to be visible when a user is in the SIG, and a user level to allow or disallow the selection of that SIG by a user. There are a number of commands available here:

Toggle Type

This switches the SIG editor back and forth between message SIGs and file SIGs. The current type is displayed in the prompt.

List SIGs

This command simply displays a list of all defined SIGs on your system.

Edit SIG

This command will allow you to modify the characteristics of a given SIG. You will be able to change the SIG's name, short name, user level, and number.

Add SIG

This command will allow you to add a SIG to your system. You will be asked to provide the SIG's name, number, short name, and user level.

Delete SIG

This command will allow you to remove a SIG from your system.

Add Area(s)

This command will allow you to add an area or several areas to a SIG's definition. Simply keep entering area numbers. A blank prompt will close the adding session.

Delete Area(s)

This command will allow you to remove an area or several areas from a SIG's definition. Simply keep entering area numbers. A blank prompt will close the deleting session.

List Areas

This command will allow you to view the areas associated with a given SIG.

DLG Command File Reference

Here is a listing of all of the supplied DLG modules, with an explanation of what each one does. Note, this chapter does not cover the command modules for DLGMail, or TrapDoor and TrapList, as there are separate chapters devoted to each of these programs. In many cases, you will not need the information in this chapter unless you are trying to assemble complex batch files or highly customize your system.

Conventions: Some of the commands will work independently of DLG, or at least independent of a DLG session. These are listed as "CLI/Batch file usable". Other commands only work in the context of an on-line DLG session (i.e. on an active port) and are listed as "Menu item usable". Those that are listed as "CLI/Batch file usable" can also be used as menu items where such a use makes sense. A third class of executable is marked as "Used internally by DLG" which means that the item is called from other DLG modules, and is not meant to be used separately, either in a menu item, in a batch file, or in a CLI.

ActivatePort

Usage: ActivatePort -p <port> [-b <background command>]

- p the port to affect
- b a background command to run when the port is not in use.

CLI/Batch file usable

Purpose: To make a given port available to the DLG system. When a port is activated, it is ready to receive calls from users. For a port to be able to be activated, it must first be mounted on your system. The Start.DLG script file in your S: directory mounts and activates all of the ports that you told the DLG installation program to setup. On a typical DLG installation, this would be port TL0 for local connections (i.e. at your keyboard) and TR0 for calls through your modem.

The background command is the full pathname of any given program that you would like DLG to run whenever the port is not actually in use by a caller. This normally defaults to "Setup", which is the DLG module that performs the task of answering the phone. If you wanted to use the TrapDoor program to answer your phone instead of DLG, then you would specify that program or a script to execute that program here. When you are activating a local port, you should use the -b option with empty quotes to indicate that NO program is to be run - the local port does not need to have Setup running and waiting for a connection.

Examples:

```
Activateport -p TL0 -b ""
```

```
Activateport -p TR1 -b "Execute Fido:Trapdoor.batch"
```

See also: Deactivateport, GetPort, FreePort, TPTRM, Setup

AddTime

Usage: AddTime -p <portname> [-m <minutes>]

- p the port to affect
- m the number of minutes to add

CLI/Batch file usable

Execute from menu as OVERLAY

Purpose: The AddTime command will allow you to add or subtract from a user's available time for that session. If the number of minutes is a negative number, that amount of time will be subtracted from the user's remaining on-line time. If you issue this command without the minutes parameter, AddTime will simply report back the number of minutes remaining in the current session.

AreaInfo

Usage: AreaInfo -a <area> -(mlf)

- a the number of the message or file area
- m or -f parameter to specify message or file areas.

CLI/Batchfile usable

Purpose: AreaInfo will return basic information about a particular message or file area. The information will be in this format:

```
Area: 1
Password: NOT LOCKED
Reason: NOT LOCKED
Priority: 0
Users In Area: 1
```

Part of DLG resource management controls access to message and file areas. An area will become locked whenever a user saves a message from the message editor, or an area will be locked if an external program needs to scan or toss messages into an area.

See also: CloseAreas, OpenAreas, TPTRM

Broadcast

Usage: BroadCast -p <port> -f <from> -m <message>

- p port to send the message to
- f name of person sending the message
- m the text of the message

CLI/Batch file usable

Execute from menu as OVERLAY

Purpose: Broadcast will allow a user on one port to send a message to a user on another port.

BroadCast requires three parameters:

-p <port> the port you wish to have the message broadcast to.

e.g. -p TR0

-f <from> the name of the person (or program) sending the message.

e.g. -f "Tom Smothers"

-m <message> - the actual text of the message (no more than 75 characters).

e.g. -m "Try out the new option on the Utilities menu!"

If BroadCast is used online, it will prompt the user for the arguments needed. If you run BroadCast from your CLI or from a batch file, all three arguments must be specified.

If you use the "*" wildcard when specifying the port name, the message will be broadcast to all active ports.

Example:

```
BroadCast -p tr0 -f SysOp -m "Hello from the SysOp!"
```

```
BroadCast -p * -f SysOp -m "Everybody into the pool!"
```

CatchUp

Usage: CatchUp -n <underscored_username> -(flm)

- n underscored username (as in Pat_Jones)
- f catchup in file areas
- m catchup in message areas

For use as a DLG menu item

Execute from menu as OVERLAY

Purpose: CatchUp will set a user's message pointers to the highest message or file in all areas they are actively reading. This is useful if a user has fallen behind in reading and wants to quickly catch-up to the current settings.

You can place this in a DLG menu using the following call string:

```
DLG: CatchUp -n %UNAME -m
```

or

```
DLG: Catchup -n %UNAME -f
```

Chat

Usage: Chat -p <portname> [-l <logfile> -s -t]

- p the port to chat with (CLI usage only)
- l the pathname of a file to record the chat session to
- s no double spacing
- t turn off clock

CLI/Batch file usable

Execute from menu as OVERLAY

Purpose: Chat will interrupt a user on a given port, no matter what they are doing (except for file transfers), and allow you to enter into an interactive conversation with them. Once you have finished chatting, either you, or the chatted user, can press CTRL-Z to end the chat session, and the user will return to the exact spot where they had been interrupted. The Chat window will appear on your Workbench screen, using the settings for windows as set up in your Global Configuration. If, for some reason, the chat window does not appear, it is likely that DLG was unable to open a window on your Workbench screen according to the setup information you provided in your display configuration.

Chat takes three arguments:

-p <port> - the name of the port that the user you wish to chat with is on. This is a required parameter if you are executing Chat in a CLI.

e.g. -p TR0

-l <logfile> - the full path name of a file to record the chat session into. This is an optional parameter.

e.g. Chat -l DHD:Jones.chat

-s - this switch disables the normal double spacing that occurs when either you or your user presses RETURN. The double spacing is useful to help differentiate the user's comments from the SysOp's. When you are using a logfile, however, you may wish to eliminate these double spaces.

-t - this switch turns off the user's clock for the duration of the chat.

When a user wishes to chat with you, they will use the Chat item from the Main Menu. When you wish to chat with a user, you will type in a CLI window "Chat -p <port>". Therefore, you will need to know the name of the port the user is on before you can chat with them. Once you are in chat mode with a user, you can call up any notes that you have made while editing that user's account by pressing CTRL-N. This is a handy way of reminding yourself who the user is, and what business you may have with them. You can create notes on a user while editing their account in the User Account Editor, available from the SysOp Menu.

When calling Chat from a menu item, you must omit the -p <port> parameter. This signals to Chat that it is being run by a user from a menu item, and causes it to issue a Page requestor rather than going directly to chat. The page requestor informs the SysOp that the user wishes to chat, and gives the SysOp the opportunity to ignore the page.

CLI Example:

```
Chat -p TR0
```

Menu Item Examples:

```
Chat -t
```

CloseAreas

Usage: CloseAreas -a "area1 area2 ..." [-k <password> -r <reason> -l <priority level> -(m|f) -b -d]

- a the numbers of the areas to close
- k the password to use
- r the reason for the closing (will be displayed online when users attempt to enter the area)
- l the priority level for the lock
- m specifies message areas
- f specifies file areas
- b borrow the area (closing the area is the default)
- d get a read lock on an area (write lock is the default)

CLI/Batch file usable

External programs need a way to lock message and file areas when they import or export messages. When an area is locked, no users can enter that area. If you lock an area that a user happens to be in at the time, the lock will wait until the user leaves that area. This command can be used to provide these necessary locks on message (-m) and file (-f) areas. Up to 256 areas can be locked with one call to the CloseAreas command. The areas to be locked are specified as a string of the form "area1 area2 ..." where area1 and area2 etc. represent either the number of an area or one of the keywords "ECHO" or "NET". "ECHO" is shorthand for all areas designated to be FidoNet EchoMail areas and "NET" is shorthand for all areas designated to be NetMail areas.

The password (-k) applies to all of the areas that you are locking and will be required to free the areas. The default password is "DLG".

While the reason (-r) argument is optional, you are encouraged to specify a reason for the locks, as this information will be displayed to a user who attempts to enter a locked area.

Area locks are managed through a priority locking scheme much like port locks (see GetPort for details). The major difference is that negative priority locks have no special significance in the area locking system. Otherwise, queuing of area locks operates in exactly the same manner as does queuing of port locks.

The resource manager provides two methods that areas can be locked. An area can be closed (-c, the default) or it can be borrowed (-b). If you close an area, not only will the CloseAreas command wait until your lock reaches the front of the queue, but it will also wait until all users have left the

area. Also, if any close locks are active or pending in an area, no users will be allowed to enter. If you borrow an area, the CloseAreas command will only wait for your lock to queue to the front; it will not wait for users to leave the area.

You should use the following criteria to determine how you should lock an area:

Closing

- Used to perform maintenance on an entire area
- Used if you will have the area locked for a long time (> 5sec)
- Use a priority of less than 64

Borrowing

- Used for time-critical activities
- Used if you will have the area locked for a short time (< 5sec)
- Use a priority of at least 64

Examples:

```
CloseAreas -a "NET ECHO" -r "Processing FidoNet/EchoMail"
```

This will close all NetMail and echo mail areas for processing.

```
CloseAreas "1 2 3 4 5" -k mylock -r "Processing NetMail" -l 0 -m
```

This will close message areas 1 to 5, with the password "mylock", at a priority of 0

See Also: OpenAreas, EnterArea, LeaveArea, TPTRM

CompileLang

Usage: CompileLang <language>

where <language> is the root name of a language to be found in DLGConfig:Misc (for example, 'English').

CLI/Batch file usable

Purpose: A language file is a text file containing every text string used in the DLG program. You can create a new language file by copying the English.lang file and editing the copy. Before a ".lang" file can be used by DLG, it has to be compiled. CompileLang is the DLG program you use to compile your language file into a form that DLG can use. CompileLang will take a language file (xxx.lang) and create a compiled version (xxx.dat).

CompileScreen

Usage: CompileScreen

CLI usable

Purpose: DLG supports message filtering in all of its message areas. Message filters are useful for preventing or discouraging the use of inappropriate or offensive language. You can define a message filter as a text file, which consists of paired words, enclosed in quotes. DLG does not directly use the text, but instead requires the text file to be compiled into binary form. This results in dramatically faster search and replace times when messages are filtered. CompileScreen is the program that you use to compile your filter files into the format that DLG needs.

To create a language filter for your system, use a text editor to create a file called "DLGConfig:Misc/Screen.MSG". This text file should contain pairs of words or phrases, in quotations. See the following example:

```
"fish" "kippers"
```

```
"cat" "feline"  
"dog" "canine"
```

Every time the string "cat" appears in a message, it will be replaced with the string "feline." DLG will capitalize the first letter of the replacement string if the first letter of the original string is capitalized. If you do not wish the translation to occur in mid-word, you must pad the word with a special delimiter character, "|". In this example, we want the word "cat" to be translated to "feline", but only when it appears as a word, all by itself. The proper way to do this is to enter the translation into the Screen.MSG file as:

```
"|cat|" "|feline|"
```

When the "|" appears at the beginning of a word in the Screen.MSG file, it means that "This is the beginning of the word" and that no other letters should appear before this string of characters. When the "|" appears at the end of a word in the Screen.MSG file, it means that "This is the end of the word" and that no other letters should appear after this string of characters. When the entire string is bracketed by "|" characters, this means that the word is to be taken as an entire word, all by itself.

Once you have set up all of your translation strings, you have to compile the translation file. You can do this by entering the command "CompileScreen" in your CLI. The first step is to "CD" to the directory where your Screen.MSG file is. In the case of the global message screen file, type the following in your CLI:

```
cd dlgconfig:misc  
CompileScreen
```

DLG will compile the Screen.MSG file into a file called "Screen.DAT" which allows for very fast message translation. You must be in the directory containing the Screen.MSG file.

ConfScr

Usage: none

For internal DLG use only

Purpose: ConfScr is an internal DLG module which operates the conferencing screen for the local SysOp.

ConfUser

Usage: ConfUser [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

For use as a DLG menu item

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: ConfUser is the DLG executable which runs PeopleLink conferencing. ConfUser presents a user with a new menu of Conferencing options, and communicates with DLG's handler for handling the complexities of conference I/O.

Example: See the default menu entry in the MAIN MENU for PeopleLink conferencing.

CronEvent

Usage: CronEvent (Add|Delete|ChangeDir|List|Read|TabList|When|Quit) [<minutes> <command>]

CLI/Batch file usable

Purpose: CronEvent is used to control TPTCron, DLG's timed event manager. When TPTCron is running, you can create a list of dynamic events with CronEvent. The events which can be scheduled with CronEvent will only be executed once, after a given time interval. CronEvent can be used to list the pending events, add events to the list, delete events from the list, or stop the TPTCron program. Here is an explanation of each of CronEvent's commands. The commands can be shortened to just the first letter of each command name:

Add (a)

When given the add command, CronEvent expects two more parameters to be given - the number of minutes to wait until executing the event, and the command to be executed. If the command to be executed has spaces or requires further command line parameters, it must be quoted. After the given number of minutes has expired, the command will be executed on the next minute mark. There is one special case for the number of minutes given to the add command - when the minutes parameter is zero, the event will occur immediately.

Syntax: CronEvent a <minutes> <command>

If the given number of minutes contains a colon (:) character, then the minutes argument is treated as an absolute time in 24 hour format. For instance, if you were to issue the command:

```
CronEvent a 9:00 <command>
```

Then the indicated command will be executed at nine o'clock AM. If this command had been issued at 8:59 AM, then TPTCron would wait 1 minute before executing the command. If this command had been issued at 9:00 AM or later, then TPTCron would execute the command at 9:00 AM the following day.

Delete (d)

When given the "delete" command, CronEvent expects one more parameter to be given - the name of the command to delete from the list. ALL occurrences of that particular command will be deleted from the list. For instance, if there were more than one DATE command in the list, then:

```
CronEvent d DATE
```

would delete all pending DATE commands. You may use ARP style wildcards with the delete command. For instance:

```
CronEvent d D*
```

would delete all pending events that start with the letter D.

```
CronEvent d ~SI
```

would delete all pending events except for SI.

Please note: CronEvent will not delete pending PERMANENT events set up through the CronTab file. You should also be aware that the comparison process used to delete events is case sensitive.

Syntax: CronEvent d <command>

Changedir (c)

When given the changedir command, CronEvent expects one more parameter to be given - the path to change to. This has the effect of changing TPTCron's current directory. All future commands launched by TPTCron will have this as their current directory.

Syntax: CronEvent c <path>

List (l)

When given the "list" command, CronEvent requires no other parameters. The complete list of pending dynamic events will be printed in the TPTCron CLI window. Please note: the list command will not list pending PERMANENT events set up through the CronTab file. If you wish to view the list of permanent events, you should use the tablist command, explained below. You will notice that when TPTCron lists the pending dynamic events, some of them will be preceded by an asterisk (*). These events have already occurred, but have not yet been removed from the list. Events are

removed the minute AFTER they have been executed. The reason this is done is that when programming with TPTCron, this list is used to see whether an event is still pending or not. Since it often takes some time for an event to take effect after it has been executed by TPTCron (i.e.: it takes time for programs to load off disk), the one minute delay between event execution and deletion provides a bit of a failsafe mechanism.

Syntax: CronEvent l

Read (r)

When given the "read" command, CronEvent can optionally be passed one more parameter - the name of a new CronTab file. The new CronTab file will replace the original table of pending PERMANENT events. If you give no pathname for a new CronTab file, CronEvent will attempt to read in the LAST CronTab file that you used. If you wish to remove all pending permanent events, but do not wish to terminate TPTCron, then you will have to create a CronTab file that has no entries, and use the read command to load that into TPTCron.

Syntax: CronEvent r [<crontab>]

Tablist (t)

When given the tablist command, CronEvent requires no other parameters. The complete list of pending PERMANENT events will be printed in the TPTCron CLI window. Please note: tablist will not list any pending dynamic events. If you wish to view the list of dynamic events, you should use the list command, explained above.

Syntax: CronEvent t

When (w)

When given the "when" command, CronEvent requires one more parameter to be given - the name of the command to list. For instance, if there was a DATE command in the list, then:

```
CronEvent w DATE
```

would print when the pending DATE command would be executed in your current CLI. If there is more than one of that type of command in the list, the "when" command will only list the first one. Note that both permanent and dynamic events are available to a When request.

Syntax: CronEvent w <command>

Note: both LIST and WHEN will list event times as the exact minute mark at which they will occur. For instance, if we enter the command:

```
CronEvent a 5 si
```

at the time 05:03:23, and then enter this command:

```
CronEvent l
```

we will see the following report:

```
TPTCron: List of pending events:
```

```
=====
```

```
Time: Tue Mar 6 05:08 - Command: <si>
```

The original seconds are ignored, and the time is rounded backwards to the even minute mark.

Quit (q)

When given the "quit" command, CronEvent requires no other parameters. The TPTCron program will be immediately shut down, and all pending events will be cancelled. Alternatively, if you run TPTCron in its own CLI, then typing a CTRL-C in that CLI will also shut down the TPTCron program. This method will cause TPTCron to abort on the next minute mark. If you wish, you can also alternately do a status command to find out what task number TPTCron has, and issue a "break <tasknumber>" command at the CLI to terminate its operation.

Syntax: CronEvent q

See Also: TPTCron

CRtoLF

Usage: CRtoLF <filename>

CLI/Batch file usable

Purpose: Amiga text files have line-feeds (LF) to indicate ends of lines or paragraphs. Text files from other types of computer may have carriage-returns (CR). CRtoLF will read a text file in, translate all of the carriage-returns to line-feeds, and save the file back out again. This command is useful when you want to edit text files from other computers.

Example: CRtoLF Macintosh.TXT

See also: LfToCR

DeActivatePort

Usage: DeActivatePort -p <port> [-i]

Where 'i' specifies Immediate mode

CLI/Batch file usable

Purpose: DeActivatePort will remove an activated port from DLG's list of available ports. A deactivated port will no longer answer the phone or accept callers. If you deactivate a port while a user is on-line, the command will pend until the user logs out. If you use the [-i] immediate mode, the port will be deactivated immediately, whether a user is on-line or not.

Example: Deactivateport -p TR0

See also: ActivatePort, GetPort, FreePort, TPTRM

DF

Usage: DF <filename>

For use as an executable in a DLG Menu Item, or in a DLG Batch File
Execute from menu as OVERLAY

Purpose: DF will display a text file. This command only works on-line. You would normally construct a menu item to display a text file. The text files can contain all of the %switches discussed in the chapter on text files. You would specify the menu item as an Executable, with the name of the executable set to: "DF <filename>". When used in this way, DF will obey the user's settings for ANSI flags and MORE prompts. You can also use DF in a script file that needs to display text, but remember that when DF is used in such a fashion, it cannot get input from the user, unless you use it in a DLGBatch file - the more prompts are disabled and any PRESS RETURN switches will be ignored.

Example: DF <filename>

DFExport

Usage: DFExport <filename>

CLI/Batch file usable

Purpose: The purpose of this command is to provide a way for you to print or generate lists or reports on all of your users. The indicated file name should be a formatted text file containing user information %switches as outlined in the chapter on text files. DfExport will go through your entire userbase and print the text file once for each user. You may redirect the output of DfExport to a disk file or your printer if you wish.

DLGEdit

Usage: DLGEdit <filename>

Drop

Purp

optio

remo

calls

and c

have chosen in their user
ing text and batch files from a
your editor preferences and
e file, DLGEdit, re-opens it
dard AmigaDos text file.

DLGUSend

Usage

Mainly

Purpos

only be

send th

the def.

and UU

sends U

directly

See Also

les (-n). This program can
ts for a UUCP address to
d with "-t" will be used as
to Newsgroup postings
by MESS when a user
to use this program

DLGXmodem

Usage: DLGXmodem [-s|lr] -c -k -q -p <protocolchar>] -f <filename>

Description of Switches:

- s: Send file
- r: Receive file
- c: Force checksum when receiving
- k: Send 1K blocks
- q: Quiet mode - do not monitor transfer
- p: Designate the one-character label for this protocol
- f: Designate the name of the file to send

Drop To DOS or DLG Protocol usage only.

Purpose: DLGXmodem is an Xmodem implementation for DLG. Xmodem is a single file transfer protocol, which has several variations. Xmodem normally sends 256 byte "blocks" of data with a checksum to ensure data integrity. Some variants of Xmodem use a more sophisticated "CRC" checksum, and others allow for larger 1K block sizes.

The -p option allows you to set a one character flag for that protocol that will appear in the title bar of the session window.

This command can only be executed within the context of an on-line DLG session. If you examine the default protocol setups in your General Configuration Editor, you will see how DLGXmodem is used. DLGXmodem can also be used from a command line in DLG's Drop to DOS mode.

Example usage in Drop To DOS:

```
DLGXmodem -s -k -f <filename>
```

See also: DLGZmodem

DLGZmodem

Usage: DLGZmodem [-(slr) -b -q -p <protocolchar>] -f <filename>

Description of Switches:

- s: Send file
- r: Receive file
- b: Batch mode
- q: Quiet mode - do not monitor transfer
- p: Designate the one-character label for this protocol
- f: Designate the name of the file to send

Drop To DOS or DLG Protocol usage only.

Purpose: DLGZmodem is a batch file transfer protocol. It is a very high-speed implementation designed for maximum throughput. It is intended to be used in the context of an on-line DLG session only. The "-b" switch can only be used in the context of a FILE initiated file transfer session - i.e. only as a DLG File Transfer Protocol option.

The -p option allows you to set a one character flag for that protocol that will appear in the title bar of the session window.

This command can only be executed within the context of an on-line DLG session. If you examine the default protocol setups in your General Configuration Editor, you will see how DLGZmodem is used. DLGZmodem can also be used from a command line in DLG's Drop to DOS mode.

Example usage in Drop To DOS:

```
DLGZmodem -sf <filename> (the "*" wildcard is allowed for multiple selections)
```

```
DLGZmodem -r (no filename is necessary, as filenames are sent with the files)
```

See Also: DLGXmodem

Door

Usage: Door ["Your Name" <Menu>]

Used only in the DLGConfig:Batch/T??Startup files

Purpose: Door is the DLG module that prompts a user for his name and password before letting him start a session with DLG. You may also optionally skip the name and password prompts for yourself on your local line by editing your DLGConfig:Batch/TLO.Startup file to replace the line that reads:

```
DLG:door
```

to read thus:

```
DLG:door "Your Name" <Menu> ;This could be MAIN or SYSOP or any custom menu
```

When you type "local" in your CLI, you will be logged directly into the BBS under your account name. This is not desirable if other people log in locally from your terminal.

EnterArea

Usage: EnterArea -a <area> -(mif)

CLI/Batch file usable

Purpose: DLG resource management requires that areas be "locked" or "borrowed" before messages or files are added to them. This command is used to enter an area as if you were a user, adding 1 to the count of users in the area. The -m or -f options indicate that the area to be entered is a message or file area. If the area is closed, EnterArea will fail. Note that unlike the CloseAreas and OpenAreas commands, EnterArea accepts only one area number. It is good practise to enter an area before you borrow it.

See Also: LeaveArea, CloseAreas, OpenAreas, TPTRM

FidoConfig

Usage: FidoConfig [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: FidoConfig is the DLG module that allows you to adjust your FidoNet settings. This is normally called from the SysOp Menu.

File

Usage: File [-a <area number> -c <command stack> -p <level> -s <sig number> -f -t -m <menuname>]

Menu Item usable

Execute from menu as CHAIN

Purpose: File is the DLG module that provides the functions of the file areas. This is normally called from the Main Menu with no command line arguments. You can also add different menu entries for FILE using different combinations of the optional command line switches. The effect of the switches is as follows:

-a <area number> - this defaults a user into a particular file area when they enter the file module.

-c <command stack> - this allows you to pre-pend a command stack onto the user's existing command stack (if he has one) in order to automate a function such as downloading a specific file. If the command stack contains spaces, it must be within quotes. In order to avoid File's auto-diversion to the user's private file directory (if he has files in there), use a tilde (~) character as the first character of the command stack.

-p <level> - this allows you to control the ability of users to send private files. The level you supply is the lowest user-level that users will be able to send files to. For example, if you set up File in your Main Menu as so:

File -p <255>

then: users would only be able to send private files to the SysOp.

-s <sig number> - this defaults a user into a particular SIG when they enter the file module.

-f - if you use the -s option to choose a default SIG when calling FILE, the -f option will force that SIG. In other words, users will not be able to change out of the SIG you gave with the -s argument when they enter the FILE base in this way. Using this feature, you can now create several different

file area SIGs available from the main menu of your BBS. For instance, your main menu could have a command "Enter AMIGA File Base", and you could call File from that menu item with the appropriate command switches. This will give the illusion of having a completely separate file base for Amiga files.

-t - using the -t switch will cause the file module to suspend the user's online timer while they are uploading a file.

-m <menuname> - allows you to specify a menu file other than the default of File_Main.

Example usage:

```
DLG:File -s <signumber> -f
```

A user entering the message base in this fashion will not be able to switch SIGs without returning to the main menu. This feature allows you to create the illusion of having several special file areas. If you leave the -f switch off the command line for File then the user will start out in the indicated SIG as a default, but will be able to change SIGs once they enter the file base.

```
File -a 1 -c ""d;DLGManual.txt;m"
```

This would force the user into file area 1, and download the file "DLGManual.txt".

```
File -s 10 -f -p 255
```

This would force the user into SIG 10, and not allow him to send private files to anyone except the SysOp.

FileAreas

Usage: FileAreas [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: FileAreas is the DLG module responsible for the creation, editing, and deletion of your DLG file areas. It is normally called from the SysOp Menu. Normally, this command is chained from the menu. Programs that chain are loaded into memory and the previous module (usually Menu) exits. Programs that overlay are loaded into memory, but the previous module stays in memory as well. When the overlayed program exits, the previous module is resumed, without needing to be re-loaded from disk. This can save a bit of time on a system that has the memory for it.

FileMaint

Usage: FileMaint [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: FileMaint is the DLG module responsible for the maintenance of your DLG file areas. In this module you can edit files, perform batch uploads of files in various ways, and "freshen" your file areas. This module is normally called both from the SysOp Menu, and also by File when you are editing files directly in your file areas. Normally, this command is chained from the SysOp menu, and overlayed when called from File. Programs that chain are loaded into memory and the previous module (usually Menu) exits. Programs that overlay are loaded into memory, but the previous

module stays in memory as well. When the overlayed program exits, the previous module is resumed, without needing to be re-loaded from disk. This can save a bit of time on a system that has the memory for it.

FilePurge

Usage: FilePurge -l <level> -n

- l specifies user level to cutoff file deletion
- n specifies that files are to be moved to File:KilledFiles rather than being deleted.

CLI/Batch file usable

Purpose: FilePurge will search all private user directories and delete files which have been downloaded successfully. Sometimes users do not delete files that have been sent to them in a timely manner, and this command should be run to clear out these old files to recover storage space. If you use the -n option, the files will be moved to File:KilledFiles instead of being directly deleted. This will allow you to inspect the files that are being transferred privately on your system, and alert you to any possible illegal transfer of copyrighted material. If you LIST the contents of File:KilledFiles, the comments on each file will provide you with the name of the person who received the file, and aid in any action you might wish to take to curtail illegal activities.

Example use:

```
FilePurge -l 100
```

- this will purge the private directories of all users with user-level 100 or less.

FileSearch

Usage: none

Internal DLG use only

Purpose: FileSearch is called by File when a user initiates a search for a filename.

FileUsers

Usage: FileUsers [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: FileUsers is the module that allows you to edit user access to your file areas. Normally called from the SysOp Menu.

FixUsers

Usage: FixUsers

CLI/Batch file usable

Purpose: DLG keeps a sorted list of users in a file called DLGConfig:Misc/Users.BBS. Sometimes this list can get out of sync with your actual user list. Running FixUsers from your CLI will re-generate this user list for you from the actual information in the User: volume. This command was more important in the early days of DLG development. Its use could be quite rare now.

Forward

Usage: none

For internal DLG use only

Purpose: This module is called by Mess when a user wishes to forward mail from either his private message area, or from a public message area.

FreeMisc

Usage: FreeMisc -n <name> [-k <password>]

-n name of the resource to free

-k the password that was used to lock the resource

CLI/Batch file usable

Purpose: FreeMisc will free a miscellaneous resource that has been locked with GetMisc. If a resource was locked with a password, you will have to specify that password to free the resource. Not all systems will have to worry about this command. It is provided as a convenience for the systems that require it.

See also: GetMisc, TPTRM

FreePort

Usage: FreePort -p <port> [-k <password>]

CLI/Batch file usable

Purpose: This command is used to free a lock on a port that has been obtained using the GetPort command. The password provided must match the password that was used to lock the port.

Example: For an example of usage, please see the example below in the discussion on GetPort.

See also: GetPort, ActivatePort, DeactivatePort, TPTRM

Freshen

Usage: Freshen

CLI/Batch file usable

Purpose: The Freshen CLI command has been provided to update your current file areas to include the new index files. Once the freshen command is used once, it is not necessary to use it again unless a third party utility (not supporting the new index file scheme) is used to place, remove or edit a file on-line. In this case, a freshen for a single file area can be performed in the FileMaint module.

GenConfig

Usage: GenConfig [-m <menuname> -o]

-m <menuname> specifies menu to load

-o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: This is the module that lets you configure "general" items in your BBS system - external editors, external file protocols, archiving utilities, etc. It is normally called from the SysOp Menu.

GetMisc

Usage: GetMisc -n <name> [-k <password> -r <reason> -l <priority> -d -i]

- n name of the resource to lock
- k the password to use when locking the resource (optional)
- r the reason for locking the resource (optional)
- l the priority for the lock (optional)
- d defaults to a write lock on the resource. If the -d option is used, it will be a read lock (optional)
- i immediate mode (optional)

CLI/Batch file usable

Purpose: When you want to prevent a batch file from being executed twice, you can use the GetMisc command to utilize TPTRM's resource tracking to "lock" the activity of the batch file, and prevent another execution of that batch file until the first execution has ended, and freed up the resource with FreeMisc. GetMisc can also be used for any general purpose resource tracking that you may want to do.

See also: FreeMisc, TPTRM.

GetPort

Usage: GetPort -p <port> [-k <password> -r <reason> -l <priority level> -b <break command> -i]

- p port to get a lock on
- k password to use in locking the port (optional)
- r reason for locking the port (optional)
- l priority level for the port lock (-128 to 127) (optional)
- b break command needed if locked with negative priority (conditional)
- i specifies immediate mode (optional)

CLI/Batch File usable

Purpose: This command is used to obtain a lock on a port. To ensure that locks are freed only by their owners, locks are password protected. To free a lock, the password that was used to lock the port must be provided. A default password of "DLG" is used if no password is specified. Passwords are not intended as a security precaution; they are provided as a measure to prevent one application from inadvertently freeing another application's locks.

Each lock can be given a string that represents the reason the lock is in effect. This string is not used internally, but it is a good practice to provide it to document your lock.

DLG manages access to ports through a priority locking scheme, with valid priorities ranging from -128 to 127. A lock request will either be satisfied immediately (if there are no other locks on the port) or it will be placed in a priority queue. Locks with higher priorities will be placed further up in the queue. Locks with negative priorities are treated specially and are considered to be "breakable" locks. When using a negative priority lock, you are required to provide a "break command" - a command that will signal your program to finish up its activities and release the lock. If a port is locked with a negative priority and a positive priority lock is issued, the break command associated with the negative priority lock will be issued. The default lock priority is zero.

Because the GetPort command will wait if its lock request is queued, a special mode has been provided. Specifying '-i' will cause LockPort to attempt to obtain an immediate lock on the port. If a lock can't be obtained immediately, the GetPort command will exit and will not wait for the lock. If GetPort was unsuccessful in immediate mode, it will return with an ErrorLevel of 5.

Example: Script file to run a terminal program on serial.device unit 0

Failat 6

```

GetPort -p TR0 -k "Comm" -r "Terminal Program" -l 1 -i
If WARN
Echo "Sorry, the serial port is busy at the moment!"
Quit
EndIf
DH0:Applications/Telecom/SuperTerm
FreePort -p TR0 -k "Comm"

```

In this example script file, you attempt to lock the port you wish to run another serial program on. In this case, you wish to run a terminal program. If the lock is unsuccessful, it is because the port is already occupied. If the lock is successful, the terminal program is started. When you exit your terminal program, the port is freed up again. TPTRM will detect that the port is idle, and execute the background process for the port, which will start DLG up again on that port. You would create similar batch files for any program that you might wish to run, but which uses the serial port (for example MIDI software). By using this form of script file to manage access to the serial ports, you can be confident that you will avoid serial port conflicts.

The only time that you cannot use this technique is with software that "detaches" from the CLI it was run from. For example if you were running Gold Disks's Professional Page, and printing to a serial printer, this technique would fail. This is because your script file would execute the GetPort command, run Professional Page, and then immediately execute the FreePort command. This is because Professional Page detaches from the script, allowing commands following it to be executed while it is still running. In this case, you would have to create a script as above to lock the port and run the software, and then run a second script file manually when you are finished and want to activate DLG again.

See also: FreePort, ActivatePort, DeactivatePort, ListPorts, TPTRM

GoodBye

Usage: Goodbye

Menu item usable

Execute from menu as CHAIN

Purpose: Goodbye is the module that is called when a user selects the "Goodbye" command from any menu in the DLG system. Goodbye gives the user a last chance to change his mind, and then bids him adeiu.

Group

Usage: Group [-m <menuname> -o]

-m <menuname> specifies menu to load

-o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: Group is the module that lets you create, edit, and delete groups. A group is a collection of users. A group can receive messages and files. When a message is sent to a group, all members of that group receive a copy of that message in their private mail areas. When a file is sent to a group, only the leader of the group — the GroupOp — will receive the file. Group is usually called from the SysOp Menu.

HangUp

For internal DLG use only

Purpose: HangUp is the module that DLG uses to make sure that a phone connection is terminated properly. The method that DLG uses to hang up the phone is user selectable - either by dropping the DTR (data terminal ready) line to the modem, or by issuing commands to the modem. The DTR method is the most reliable of the two, but is not always available or desired.

Immed

Usage: Immed <port> <lockbaud> <realbaud> [-w]

CLI/Batch file usable

Purpose: Immed is usually called by a front door program when an incoming call is detected. Immed executes a script file found in DLGConfig:Batch. Each port has a different, individual script file. The script files are named <portname>.startup. For example, the port "TR0" would have a script file called "TR0.Startup" in DLGConfig:Batch. These script files define the DLG session.

Immed can also be called by script files, and by programs other than Setup which you might use to answer the phone. For example, Trapdoor uses Immed to pass a human caller onto the DLG system.

Immed takes some parameters:

<port> this is the three letter name of the port you want to initiate a DLG session on.

<lockbaud> this is the data rate connection between the computer and the modem. This can be different from the modem to modem connect rate, especially with modems that support RTS/CTS handshaking.

<realbaud> this is the data rate of the connection between the two modems (yours and the remote's).

-w this is an optional parameter. It causes Immed to wait until the end of the DLG session. This can be necessary if you are using a front-door program that answers the phone and passes the call off to DLG if it is a human caller. Normally, the front door program will wait in the background until the BBS quits. Since DLG is a modular system, no BBS "program" per se is running all the time. Immed runs very quickly, establishing the connection between DLG and the outside world, and usually exits right away. The front-door program would normally see this and "think" that the BBS session was finished. The -w option causes the Immed command to execute and then wait until the BBS session is complete, thus "fooling" the front-door program into holding off until DLG is finished with the call.

Examples: If you list the LOCAL script command in your S: directory, you will see that it consists of a single line:

```
Immed TL0 19200 19200
```

This starts up the DLGConfig:Batch/TL0.Startup script file, and establishes a local connection to DLG through the local port, at a data connection rate of 19200 baud. In this case, the <realbaud> parameter is a dummy value, since no actual modem connection is taking place.

If you are using Trapdoor to answer incoming calls, instead of Setup, then in your Mail:Trapdoor.cfg file you would have a line that reads:

```
BBSCOMMAND "DLG:Immed TR0 %b %B w"
```

The "%b" and "%B" parameters are switches that Trapdoor uses to pass the <lockbaud> and <realbaud> parameters to Immed. Note that Trapdoor does NOT require the use of the optional wait [w] parameter. Trapdoor exits once it has successfully passed the call over to DLG. DLG's resource manager can be configured to automatically re-activate Trapdoor once the call is complete.

See also: Setup, Door, TPTRM

Jive

Usage: Jive [textfile]

CLI/Batchfile usable

Purpose: Jive is a non-Telepro product. It translates a text file into a slang dialect known as "jive". It is included with the DLG product to demonstrate the possibilities that message processing can give you. Normally, a text file displayed with JIVE will be sent to the standard output. If you wish, you can use redirection to send the output to a text file. This is how the script file called DLGConfig:Batch/Spare1 converts the contents of a user-written message into a jive message before the message is saved in the message area.

See also: Kraut, ValSpeak

Kraut

Usage: Kraut [textfile]

CLI/Batchfile usable

Purpose: Kraut is a non-Telepro product. It translates an English text file so that it "sounds" like the writer has a German accent. It is included with the DLG product to demonstrate the possibilities that message processing can give. Normally, a text file displayed with KRAUT will be sent to the standard output. If you wish, you can use redirection to send the output to a text file. This is how the script file called DLGConfig:Batch/Spare2 converts the contents of a user-written message into a "kraut" message before the message is saved in the message area.

See also: Jive, ValSpeak

LeaveArea

Usage: LeaveArea -a <area> -(mlf)

<area> is the number of the area to leave

(mlf) denotes the area as either a message (-m) or file area (-f)

CLI/Batchfile usable

Purpose: Leave an area that was entered with EnterArea, and decrement the number of users in that area by one. No password is used, so ensure that the area being left is the same one that you entered!

See also: EnterArea, CloseAreas, OpenAreas, TPTRM

Lex

Usage: Lex <Filename> [Filename2...]

CLI/Batch file usable

Purpose: Perform a readability test on a text file. The test is neither rigorous nor very useful, but it can be a bit of fun.

LFtoCR

Usage: LFtoCR <Filename>

CLI/Batch file usable

Purpose: Translates a text file from Amiga format (lines ending with line feed characters) to a generic text format (lines ending with carriage returns).

See also: CRtoLF

LineEdit

Usage: LineEdit <BodyFile> <FileSize>

where:

<BodyFile> is the filename of the message

<FileSize> is the maximum allowable size of the message in bytes

Used as an on-line editor in the context of a DLG session

Purpose: This is DLG's line-oriented message editor. It functions more simply than does the full screen editor supplied with DLG, and is useful for users who have less sophisticated terminal programs. DLG's line editor differs from line editors in other BBS programs in that input is free-form. You may type the message continuously, and backspace past the beginning of a line to re-edit material on the line above. When this happens, LineEdit re-sends the current paragraph with the cursor situated at the end of the previous line. This is done to avoid using ANSI screen positioning character sequences, which some terminal programs do not support. Simple search and replace editing features are supported. If the user has a level of 255 (SysOp), LineEdit has an option to load and save the text buffer to and from disk. When loading text into the buffer, it will be appended to the current buffer.

ListPorts

Usage: ListPorts [-k <password>]

CLI/Batch file usable

Purpose: Lists all ports with active locks with a given password. If no password is provided, all ports will be listed.

See also: GetPort, FreePort, ActivatePort, DeactivatePort, TPTRM

ListUser

Usage: ListUser

Menu item usable

Execute from menu as OVERLAY

Purpose: ListUser will list the users on your system. It is designed to work in the context of an on-line session, and so it will not work in a CLI or in Drop to DOS mode. ListUser presents the user with a prompt for a search string to limit the search to a subset of the entire user list. Wildcards are used - ? indicates a match for any character, and * indicates a match for any string.

MailPack

Usage:

Online Mode (Menu Item): MailPack -t <msg.threshold> -b <background.pri> [-m <max.msgs>]

Execute from menu as OVERLAY

CLI Mode: MailPack -n <name> -w <width> -a <arc> -g <(rld)> -b <pri> -l <line terminator> -o

Where:

- t Threshold value. If the number of message to be packed is below the threshold, they will be packed in RAM:, otherwise they are packed in FILE:.
- b The priority of the task while it is packing messages.
- m (optional) The maximum number of messages that will be processed at a time. If not specified, there is no limit.
- n The UNDERSCORED name of the user that mail is packed for.
- w The screen width the mail will be packed for.
- a The archiving method used in packing. This number corresponds directly to the archivers specified in the user options menu.
- g Required flag to let MailPack know it is running in CLI mode.
- rld Either pack in RAM (r) or on disk (d).
- l The line termination to be used.
 - 1 Terminate with linefeeds.
 - 2 Terminate with carriage returns.
 - 3 Terminate with CR/LF.
- o If on, old mail packets will be deleted. If not, and there are old packets remaining, MailPack will abort.

Purpose: MailPack packs messages from message areas that the user has selected in their global message areas into an archive which the user will be able to download from their private file area. The messages are translated to standard text files, and packed using the user's preference archiver. When MailPack is used from a menu item, the program enters an interactive mode where the user can direct what will be packed and how. Mail is then packed in the background, while the user can go about doing other things on the system. You can also elect to use MailPack as an executable from TPTCron, to pack mail for users at a certain time of day, before they call in. We do not recommend a priority setting outside of the range from -1 to 1.

MakeFList

Usage: MakeFList <listfile> [<introfile> <tagalongfile>]

where <listfile> is the name of the file to produce. Standard FidoNet convention calls for FILES.TXT, but any name may be used.

<introfile> is the name of an optional text file which will be prepended to the beginning of the file created by this program.

<tagalongfile> allows you to send a file along with each file request.

CLI/Batch file usable

Purpose: A "file-request" is a FidoNet convention whereby one system can request that another system send files from its file areas. The files that are available for this type of transfer are listed in a text file, usually called FILES.TXT. When you are running a file-request aware front-end program like Trapdoor, a system can call in and make a file request for this list by using the "magic" filename FILES. Magic filenames can be defined for other commonly requested files on your system. A magic filename is a convenient short-form or mnemonic name that you can define for any file on your system. MakeFList generates a list of files which may be file requested from your system. It creates a text file containing the filenames of files that are not password protected. If DLGConfig:Misc/TPTFreq.LST contains a list of magic filenames, the filenames will be listed as well. MakeFList will not show files that are password protected OR in file areas not designated by you as file requestable. MakeFList timestamps the file produced to show the date it was compiled.

You can specify descriptions for magic filenames up to a maximum length of 50 characters. The description must be on a line following the entry, starting with a ':'

```
DH0:Test !Password
:This is a file called test...
```

Menu

Usage: Menu

Menu item usable

Execute from menu as CHAIN

Purpose: Menu is the DLG module responsible for running and maintaining user created menus - i.e. menus that are not tied to any one particular module, like Mess or File. An example of a Menu controlled menu would be the Main Menu.

Mess

Usage: Mess -a <area> -c <command stack> -s <signumber> -f -m <menuname> -p

Where:

- a <area> will take a user to the given message area, provided they don't have private mail waiting.
- c <command stack> will prepend the given command stack to the start of any command stack the user might currently have. The command stack will direct the user into a particular course of action. A tilde character (~) at the start of a command stack for Mess will cause Mess to not look for private mail when activated.
- s <signumber> will put the user into the given SIG when they enter the message base.
- f this command line option will "lock" or "force" the user to stay in the given SIG while they are in the message base. You can use this in conjunction with the -s <signumber> option to provide the illusion of having several different special message bases from the main menu. This option basically disables the SIG command in Mess.
- m <menuname> this will cause Mess to load a menu other than its default of MSG_Main.
- p <level> this indicates a private send level. Either the sender or the receiver must have a user level equal to or greater than the level specified here in order to send a private message. For example, if you set the -p option to a level of 255, then anyone would be able to send private messages to the SysOp and the SysOp would be able to send private messages to anyone, but none of the users would be able to send private messages to each other.

Menu item usable

Execute from menu as CHAIN

Purpose: Mess is the module that provides all of the services of the DLG message areas. For more information on Mess' role in the DLG system, please see the chapter devoted to it.

MSGAreas

Usage: MSGAreas [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: This module is the on-line SysOp message area configuration program. It provides the services necessary for the creation, deletion, and maintenance of DLG message areas.

MSGUsers

Usage: MSGUsers [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: This module is the on-line SysOp message area user access editor. It provides the services necessary for adding or deleting users from various message areas, or changing user access within a specific message area.

NewUser

Usage: none

Used internally by DLG

Purpose: NewUser is the module which administers the new user questionnaire, displays NewUser.txt, and executes NewUser.(dlg)batch.

NodeInfo

Usage NodeInfo <nodelist path>

Menu item usable

Execute from menu as OVERLAY

Purpose: NodeInfo displays info derived from a FidoNet nodelist, specifically one that has been processed by the program TrapList. NodeList enters the user into an interactive session where they are prompted for Zone, Net, and Node information. The module then displays information about that node.

OpenAreas

Usage: OpenAreas -a "area1 area2..." [-k <password> -(mlf)]

where:

- a A list of up to 256 message or file areas (by number) or one of two keywords: ECHO - All EchoMail areas, or NET - All NetMail areas
- k the password that the areas were locked with.
- mlf used to indicate either -m for message areas, or -f for file areas.

CLI/Batch file usable

Purpose: This command is used to free the locks on message or file areas that were closed with the CloseAreas command.

See Also: CloseAreas, EnterArea, LeaveArea, TPTRM

Port

Usage: Port [-m <menuname> -o]

-m <menuname> specifies menu to load
-o specifies overlay mode for the command

Menu item usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: Port is the on-line SysOp port editor. It provides the services necessary to configure all of the attributes of DLG ports, including display, modem, and general configuration items.

PortInfo

Usage: PortInfo -p <port>

-p specifies the port you wish to examine

CLI/Batch file usable

Purpose: This is used to obtain information about the active lock on a port. Here is a sample of the output:

```
Port:          TR0
Password:      Setup
Reason:        Setup - Waiting for call
Priority:       -1
Break Command: DLG:FreePort TR0
```

PrepAreas

Usage: PrepAreas

CLI/Batch file usable

Purpose: Some utilities like ConfMail will not write messages properly into message areas in which the last message was deleted. This will have an effect on the high-message pointers for users in that area. To avoid this problem, PrepAreas scans each message area and fixes all of the pointers so that the problem will not occur. It is a good idea to run this command in your batch file just prior to running ConfMail. This command automatically prepares all echo and NetMail areas.

NOTE: Do not use the CloseAreas command to get locks on the message areas before using this command — PrepAreas does all the area locking on its own.

Quote

Usage: <QuoteChar> <TextToQuote> <ReplyTo> <Width>

QuoteChar is the character that is to precede quoted text - usually ">". The quote character should not be enclosed in quotation marks on the command line.

TextToQuote is the file which will be quoted

ReplyTo is the name of the person being quoted (i.e. it will appear as "In a message dated xxx, <ReplyTo> said...")

Width is the screenwidth to use when word-wrapping the quoted text.

CLI/Batch file usable

Purpose: Quote will take an existing DLG message and create a quote file for use with an external editor. Here is an example DLGConfig:Batch/LocExtEditor batch file that uses the Quote command:

```
.key replyto,messagename
.bra [
```



```
.ket ]
cd wp:
if exists [replyto]
  dlg:Quote > [messagename] [replyto] 70
  ; note that the ">" is the quote character, not a file redirection!
endif

if exists [messagename]
  dlg:crtolf [messagename]
  WordPerfect [messagename]
  dlg:lftocr [messagename]
  quit
endif

WordPerfect
dlg:lftocr [messagename]
```

This will automatically quote the entire contents of any message you are replying to with an external editor.

ReadLog

Usage: ReadLog

Menu Item Usable

Execute from menu as OVERLAY

Purpose: ReadLog is an executable that you can add to a menu so that your users can see the activity logs of your BBS for the last seven days. When a user selects the menu entry that uses ReadLog, they will be prompted for which day they would like to see the log for. ReadLog filters the log display, eliminating items that exceed the user's level.

RemovePort

Usage: RemovePort -p <port>

CLI/Batch file usable

Purpose: This command will deactivate a port. A deactivated port will no longer accept calls until the port is activated again. If a user happens to be on the port when the command is issued, the user's session will be terminated. It is not a good idea to use RemovePort when an active session is taking place on the port. This program does the equivalent of:

```
TFlags -p <port> -K
```

```
TKill -p <port>
```

If you wish to terminate a user's session, use the TKill command alone, since it will pend until the port is running a process that can be killed.

See Also: ActivatePort, TFlags, TKill

RemoveUser

Usage: RemoveUser <port>

CLI/Batch file usable

Purpose: This command will log a user off the chosen port if the user has exceeded his maximum session time. RemoveUser sends the text file DLGConfig:Text/2MinutesWarning.txt two minutes before the deadline. It will then send the text file DLGConfig:Text/0MinuteWarning.txt just before logging the user out.

If you wish to terminate a user's session immediately, use the TKill command. This, however, will not print any warning messages, and will still wait until the port is running a process that has killing enabled. When invoked from a CLI, RemoveUser will remove any RemoveUser CronEvents for the given port. This does two things: 1) the %TLCALL switch will immediately start displaying the 2 minute warning, and 2) prevents the possibility of the user's remaining time counting down to two minutes and TPTCron sending the RemoveUser command a second time. (A RemoveUser command is entered into TPTCron's event list when a user logs in.)

See Also: TFlags, TKill

Renumber

Usage: Renumber [-a "area1, area2, ..." -m]f -r]

-a (optional) Either a list of areas to renumber, or the the following keywords:

ECHO - all EchoMail areas

NET - all NetMail areas

LOCAL - all local areas

The default for message areas is ECHO NET LOCAL, and for file areas is LOCAL

-m]f (optional) Switch. Denotes either message or file areas. The default is message areas.

-r (optional) If this switch is set, all message areas specified will be renumbered. If it is not set, then only those message or file areas that have their number of messages or files at or above the specified threshold level will be renumbered.

CLI/Batch file usable

Purpose: Renumber a message or file area. Normally a message or file area will be renumbered only when the threshold defined for that area is reached or exceeded.

ResourceReport

Usage: ResourceReport [-p <port> -m -f -r -l]

- p the name of the port to report on
- m report on message area resources
- f report on file area resources
- r report on miscellaneous resources
- l report on associated locks

CLI/Batch file usable

Purpose: ResourceReport will give you an overview of the resources that TPTRM is currently monitoring. You can selectively ask ResourceReport to tell you about specific resources. If none of the options are selected, ResourceReport shows you all of the resources.

ScreenEdit

Usage: ScreenEdit ~b ~h ~r %SCWIDTH %SLENGTH ibmfont %ANSI

Usage: ScreenEdit <bodyfile> <headerfile> <quotefile> <width> <height> [stdliolo <iostream>]
[helpfile <helpfile>] [type <msgtype>] [amigafontlibmfont] [colorlmono] [twitlSysOp] [7bitl8bit]
[quote] [save <pathname>]

default: stdio amigafont color twit 7bit type <headerfile.timesread> helpfile dlgconfig:misc/
screenedit.help minimum width 77, height 6

Configurable Editor for use with the Configuration Editor

Purpose: ScreenEdit is a flexible full screen editor for use locally and on-line. ScreenEdit has many command-line parameters to activate or deactivate its various features. To make ScreenEdit available to your users, you will need to use the SysOp General Configuration editor to add an entry for each type of screen editor environment that you wish to offer. Here are some example configurations that you might want to add to your DLG system:

User Amiga ScreenEdit with save feature:

```
ScreenEdit "b" "h" "r" %SCWIDTH %SLENGTH %ANSI SAVE User:%UNAME/Message.TMP
```

Amiga User without save feature:

```
ScreenEdit "b" "h" "r" %SCWIDTH %SLENGTH %ANSI
```

IBM User with save feature:

```
ScreenEdit "b" "h" "r" %SCWIDTH %SLENGTH ibmfont %ANSI SAVE User:%UNAME/Message.TMP
```

IBM User without save feature:

```
ScreenEdit "b" "h" "r" %SCWIDTH %SLENGTH ibmfont %ANSI
```

Amiga SysOp ScreenEdit

```
ScreenEdit "b" "h" "r" %SCWIDTH %SLENGTH %ANSI SysOp
```

To use a given editor, it must be chosen in the user options. While in the editor you have a number of commands to use while editing text. Here are some of the available commands:

ESC-S : Search ESC-R : Search/Replace

ESC-G : Get File (SysOp) ESC-W : Write File (SysOp)

ESC-? : Display a listing of available editor commands.

There is a feature to the full screen editor so it can allow regular users to save and retrieve a text file. For users with non-SysOp access to the editor, this filename is hard-coded. To set this up, add the following to the end of your full screen editor call string: Save User:%UNAME/Message.tmp

The above example would cause any temp files to be saved to the users private directory. If a user is out of time while composing a message, he can save a partial message to disk, and pick it back up on his next call to the system.

SendMsg

Usage: SendMsg [-f <from> -s <subject> -b <body>] -r <route> -n -q -e

CLI/Batch file usable

Execute from menu as OVERLAY

Purpose: SendMsg is a multi-purpose message writing utility. It allows for interactive or automatic posting of messages to one or more destinations. It is very useful when you wish to set up an automatic posting (such as the rules of a message base or reminders of monthly meetings) and when you want to send the same message to a number of different people.

Valid route methods are:

<name> - Route message privately to user or group <name>

AREA <area> <name> - Message in <area> to <name>

NET [-cfap] <addr> <name> - NetMail to <name> at <addr> with NetMail flags:
c=crash f=freq a=attach p=public UUCP <addr>

UUCP message to <addr>

USENET <newsgroups> - UseNet message in groups <newsgroups>

FILE <filename> Read one or more routing methods from <filename>

Note: The <route> argument must be enclosed in quotes if it is more than one word

The 'FILE' route method can be used recursively, and you can nest files of routing methods as deep as your stack will allow (overhead is about .5K per level). If SendMsg is called interactively (ie. it is run from a menu while a user is on-line), all arguments except the route file can be omitted. If the <from> argument is omitted, the user's name will be used. If the <subject> argument is omitted, it will be prompted for. If the <body> argument is omitted, the user's selected editor will be invoked to enter the body text.

If '-n' is specified, SendMsg will not attempt to put the message in paragraph form. Line feeds will be left exactly as they are. Without '-n', sendmsg will ignore all line-feeds that are not followed by a non-alphanumeric character. This general gives a good paragraph format from text composed in an editor that places line-feeds after every line.

If '-q' is specified, SendMsg will operate in "quiet" mode. No output from SendMsg will be shown if the '-q' option is used.

If "-e" is specified, SendMsg will load the editor. If a bodyfile is specified and the '-e' option is used, the bodyfile will be loaded into the editor.

SendMsg can easily be used to add a reliable Feedback to SysOp menu item. For example, you could put a menu command for Feedback, and set the executable to:

SendMsg -f "%NAME" -s "Feedback" -r "SysOp"

Any changes the user makes to the header information while in the editor will not take effect as this would alter the intent of the SendMsg call as set up by the SysOp.

SetMenuPri

Usage: SetMenuPri [<value>]

CLI/Batch file usable

Purpose: SetMenuPri will change the priority settings on all of your customized menu items. If you do not specify a priority, then all menu items will be set to a default priority, which can change from port to port, depending on your port configuration. If you specify a value here, then all menu items will be set to have that priority. SetMenuPri will affect all of the menus you have defined on your system.

Setup

Usage: Setup <port>

CLI/Batch file usable

Purpose: This program configures the modem, and sits and waits for a call. When a call comes in, Setup then passes the port along to the Door program which queries the user for his name and password. Setup will treat "CONNECT FAST" as "CONNECT 19200" to provide compatibility with Telebit modems.

Example: See any of the DLGConfig:Batch/TRx.Startup files to see how Setup is used.

Sigs

Usage: Sigs [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu Item Usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: Sigs is the on-line SysOp SIG (Special Interest Group) configuration editor. It must be run in the context of an on-line session.

SysMenu

Usage: SysMenu

Menu Item Usable

Execute from menu as CHAIN

Purpose: SysMenu is the on-line SysOp menu configuration editor. It must be run in the context of an on-line session. Note: This is the only SysOp editor that does not allow for a configurable menu or the overlay option.

SysUser

Usage: SysUser [-m <menuname> -o]

- m <menuname> specifies menu to load
- o specifies overlay mode for the command

Menu Item Usable

Execute from menu as CHAIN or with OVERLAY option (-o)

Purpose: SysUser is the on-line SysOp user account editor. It must be run in the context of an on-line session.

TBaud

Usage: TBaud -p <port> -b <baud>

- p the port to affect
- b the baud rate to set the port to

CLI/Batch file usable

Purpose: TBaud can change the baud rate for a specified port. Note that this is not usually something you will need to do, but is provided for special circumstances.

TColors

Usage: TColors -p <port> -c <colorlist>

- p the port whose screen you would like to affect
- c A list of colours in hex notation, in the form 0x000, where the last three digits can have any value from 000 to FFF, and define the values of the RGB registers for each of the palette colours

CLI/Batch file usable

Purpose: TColors can set the palette for a custom screen. The number of colours you specify must be appropriate for the type of screen opened. Note that a screen must be open for this command to have any effect, and the settings are not saved. This command is somewhat obsolete, and is not being actively supported. Custom screen palette colours can now be defined using the Port Configuration Editor.

TCont

Usage: TCont -p <port>

-p the port to restore

CLI/Batch file usable

Purpose: TCont will restore I/O to a port that has been frozen with the TFreeze command.

See also: TFreeze

TDevQuery

Usage: TDevQuery -p <port>

CLI/Batch file usable

Purpose: TDevQuery will display the device, unit number, and serial flags for the given port

TDIH

Usage: TDIH

CLI/Batch file usable

Purpose: TDIH is the "Today In History" program. It expects to find the "month" files to be in DLGConfig:Text/. For more information on the TDIH month files, please see the chapter on text files.

TFlags

Usage: TFlags -p <port> [-Bib Clc Did Ele Kik Wlw Xlx Sls Rlr Vlv Tlf]

- p the port to affect with the following flag settings. Uppercase flags enable the particular option, and lowercase flags disable that option.
- Bib Send breaks.
- Clc Do carriage return to line feed conversions.
- Did Pass control-D's.
- Ele Echo characters back to the user.
- Flf Freeze the output when a user starts typing in line mode.
- Kik Enable killing.
- Wlw Keep track of pending kills.
- Xlx Turn on pass-through mode.
- Sls Enable pausing.
- Vlv Enable verbose pausing.
- Rlr Turn on raw mode (otherwise the port is in line mode).

-Tlt Enable timeouts.

CLI/Batch file usable

Purpose: TFlags can enable or disable various settings in the TPTHandler for each port. The settings of the flags will affect how the handler passes data through that port, how it interprets various control codes, what type of CLI environment to emulate, and other options. You may need to use the TFlags command in batch files that run certain types of on-line games, or to enable or disable the ability of DLG to be able to kill that port while there is an active session.

Example:

To enable killing of the port (this will NOT kill the port, just make it possible):

```
TFlags -p <port> -K
```

To enable a RAW CLI mode:

```
TFlags -p <port> -R
```

To disable user account timeouts:

```
TFlags -p <port> -t
```

To disable killing of the port, but keep any kill messages for when the port becomes "killable" again:

```
TFlags -p <port> -k -W.
```

With this setting, if the port receives a kill message, but the port has killing disabled, then the port will be killed the moment that killing is re-enabled, as it will "remember" the received kill message.

See also: TKill

TFreeze

Usage: TFreeze -p <port>

CLI/Batch file usable

Purpose: Suspend all I/O on the given port. TCont will allow the I/O on the port to continue.

See also: TCont

TKill

Usage: TKill -p <port>

CLI/Batch file usable

Purpose: Shut down the handler on the specified port. Killing must be enabled for the TKill command to work. A port killed with the TKill command is still active, and will continue to receive calls. TKill just stops the current session on the given port.

TPTBC

Usage: none

Used internally by DLG

Purpose: TPTBC is a background process used by the DLG system to manage messages that are broadcast to on-line users. This is loaded in the DLG.Start file.

Tptconf

Usage: none

Used internally by DLG

Purpose: TPTConf is a background process used by the DLG system to manage communication between ports, as in conferences or on-line games. This is automatically loaded when the conference is first used.

TPTCron

Usage: TPTCron [-t <crontab> -l <logfile> -b <output>]

- t optional parameter for a list of permanent periodic events to execute
- l optional parameter to log activity to a file
- b optional parameter to send the output of TPTCron to a file

CLI/Batch file usable

Purpose: TPTCron executes AmigaDos tasks at regularly scheduled times. TPTCron is capable of executing periodic events - tasks which must be executed at specific time intervals. It is also capable of executing dynamic, one-time events, scheduled through the use of an external command called CronEvent. These dynamic events can be scheduled "on-the-fly" once TPTCron has been started up.

When TPTCron is first run it optionally reads a table of permanent, periodic events to be run from a file called CronTab, which is generally located in the S: directory. TPTCron maintains this list of events to execute in RAM.

TPTCron should be run in its own CLI window for two reasons:

- (1) The AmigaDOS commands that TPTCron executes need a place to print output, display errors, etc.
- (2) If you run TPTCron from your main CLI window, you could prevent a cron event from being executed properly by typing into that window.

If you do not wish to run TPTCron in its own CLI window you can RUN it from your main CLI by re-directing it's output to NIL:, or more preferably, NULL:. This will make it impossible to hold up a cron event by typing commands into that CLI, but it will also make it impossible to see any output from TPTCron or the programs it runs.

To re-direct TPTCron's output to NIL:, in case you wish to run TPTCron from your main CLI, the syntax is as follows:

```
TPTCron >NIL: [-t <crontab> -l <logfile> -b <output>]
```

To run TPTCron with the ability to execute periodic events, you must first use a text editor to create a "CronTab" file. The format of CronTab is very simple - it contains entries in the form of lines where each line has 6 fields and each field is separated by "white space" (either tabs or spaces) from it's neighbor. The fields are as follows:

Field	Name	Range of legal values
1	Minute	0 - 59
2	Hour	0 - 23 (0 = 12:00 a.m. / 23 = 11:00 p.m.)
3	Date	1 - 31
4	Month	1 - 12 (1 = January / 12 = December)
5	Day	0 - 6 (0 = Sunday / 6 = Saturday)
6	Command	This is the command to be run at the appointed time. It will be run just as if typed into the CLI.

Each entry for each of the first five fields should be a number in the legal range for that field. If you substitute an asterisk (*) for a number in a field it is taken to mean ALL possible numbers for that field. You may also specify a SET of values for each field by separating them with commas. Similarly, you can specify a RANGE of values by separating them with hyphens (-). For example:

Print the date in the TPTCron CLI window every minute:

```
* * * * * DATE
```

Since there are asterisks in each field, TPTCron interprets that to mean that at each legal interval, run the command DATE. Since the smallest interval is a minute, the DATE command will be executed every minute, regardless of the hour, date, month, or day.

CronTab entries MUST be left-justified starting in column 1 and each entry must contain 6 fields, each separated by spaces or tabs. There can be as many entries in the CronTab table as you like.

You may also put comments into your CronTab file, with the use of the "#" character. If the first column contains a "#", the line is considered to be a comment, and the rest of the line will be ignored by TPTCron.

The events that TPTCron schedules from the CronTab file are called PERMANENT events. If you wish to manipulate the permanent events once TPTCron is running, you must re-edit your CronTab file, or have a secondary CronTab file available, and use the CronEvent program (see CronEvent) to read the new CronTab file into TPTCron.

You can talk to TPTCron via ARexx. The ARexx port is called "tptcron.control" and it supports the following commands:

ADDEVENT <time> <command>	- Like CronEvent Add
DELEVENT <command>	- Like CronEvent Delete
CHANGEDIR <path>	- Like CronEvent Chgedir
WHENEVENT <command>	- Like CronEvent When
READFILE <crontab>	- Like CronEvent Read
CRONEXIT	- Like CronEvent Quit

The primary result codes are as follows:

If no result string has been requested, TPTCron will return RC_OK (0) if the command was executed successfully. If an error occurred, RC_WARN (5) is returned. If a result string was requested, TPTCron will return "Command successful" or an appropriate error message.

There are two exceptions to the above error code. On a DELEVENT command, the result code will be the number of events deleted. On a WHENEVENT command, the result code contains the time that the next event matching the specified time will occur (encoded in the Amiga time format - number of seconds since 1-Jan-1978 00:00) or 0 if no event was matched. If a result string was requested, the time will be expressed as a string.

See also: CronEvent

TPTFreq

Usage: TPTFreq <badfreq> <infile> <outfile>

<badfreq> the name of the file to be sent if an invalid file request is received

<infile> the name of the .REQ packet to be processed. This is to be supplied by the front-end software. For example, if you were using Trapdoor as your front-end program, then you would set infile to be %i in Trapdoor's configuration file. Trapdoor would then expand the %i switch into the file name of the .REQ packet.

<outfile> The name of the .RLO file to be produced. Once again, the front end software would supply this name. As with the example above, set this to %o in Trapdoor's configuration, and it will expand it to the file name to be used when it calls TPTFreq.

Usable from a front-end software package, such as Trapdoor.

Purpose: TPTFreq is DLG's file request server. It works with a front-end program such as Trapdoor to deliver a requested file to a remote system.

See also: MakeFlist

TPTQuote

Usage: TPTQuote <quotefile>

<quotefile> the filename of a quote file to be found in the DLGConfig:Text/ directory

CLI/Batch file usable

Purpose: This program generates a quote from the quote file located in DLGConfig:Text/. Please see the chapter on text files for more information on the format of the quotation files used by TPTQuote.

TPTRM

Usage: None

Used internally by DLG

Purpose: TPTRM is DLG's resource manager. It mediates access to ports, message areas, and file areas through a system of locks and passwords. This is automatically loaded when a port is first activated.

See also: OpenArea, EnterArea, LeaveArea, GetPort, FreePort

TPTShell

Usage: TPTShell

Menu Item usable

Execute from menu as CHAIN

Purpose: This is the Drop to Dos command available from the default Utility menu. It requires the file DLGConfig:Misc/TPTShell.cfg to function. It uses the contents of that file to determine what commands are available to users. For a complete discussion of the format of the TPTShell.cfg file, see the chapter on text files. The SysOp has no limitations on the commands that are available. Note that many CLI commands that were disk-based under Workbench 1.3 are now part of the Workbench 2.x shell, and as such are unavailable to TPTShell. If you are running Workbench 2.x, you will need to copy these commands from a Workbench 1.3 disk into your C: directory. Commonly used commands that were made part of Workbench 2.x's Shell are CD and ECHO.

TransferPort

Usage: TransferPort -p <port> [-k <password> -x <newpassword> -r <reason> -l <priority> -b <break command>]

- p the port to affect
- k the old password used on the lock (optional unless the new password option is given)
- x the new password to use on the lock (optional)

- r the reason for the lock on the port (optional)
- l the priority of the lock (optional)
- b the command to use to release the lock on the port (optional)

CLI/Batch file usable

Purpose: TransferPort can be used to transfer or modify the lock on a port. The command is very similar in use to GetLock, except that TransferLock will modify an existing lock on a port, rather than create a new one. If you are changing the password on a lock, you must supply both the old and the new passwords in the command line.

TRecover

Usage: TRecover -p <port>

CLI/Batch file usable

Purpose: If a port becomes inadvertently frozen due to failure of a CTRL-C and EndCLI sequence, TRecover can be used to reactivate the handler on that port.

TScreen

Usage: TScreen -p <port> -(olc) [-w <width> -h <height> -d <depth> -(Rlr) -(lli) -f -s <size>]

- p the port to open or close a screen on
- olc open or close the screen on the given port
- w the width, in pixels, of the screen to open (optional)
- h the height, in pixels, of the screen to open (optional)
- d the depth of the screen in bitplanes (1-3)
- Rlr use R for hi-res, and r for low-res (optional)
- lli use l for interlace, and i for non-interlace (optional)
- f the name of the font to use on the screen. This is case sensitive, and must match the case of the font name as it is found in your fonts: directory (optional)
- s the size of the font to use. (optional)

CLI/Batch file usable

Purpose: TScreen can open a custom screen to let you observe the activity on the specified port. The same command is used to both open and close a screen. Note that no reality checking is done to see that the width, height, hi-res and interlace options add up to a possible screen size. If you do not give any of the optional parameters, DLG will use the configuration settings for the given port.

See also: TWindow

TStat

Usage: TStat -p <port>

CLI/Batch file usable

Purpose: Provides debugging information on a port's status in terms of its binary encoded handler flags. If you contact TPT for technical assistance, you may be asked to provide the information that TStat can give you.

TString

Usage: TString -p <port> -s <string>

CLI/Batch file usable

Purpose: TString can be used to simulate user input on the given port. The given string must be enclosed in quotation marks if it contains spaces.

TTimeDelay

Usage: TTimeDelay -p <port> -d <delay>

- p the name of the port to affect
- d the number of 5 second intervals to add to the port's idle timeout value

CLI/Batch file usable

Purpose: TTimeDelay can set the idle timeout delay for the given port in increments of 5 seconds.

TTitle

Usage: TTitle -p <port> -t <title>

- p the port to affect
- t the text of the title

CLI/Batch file usable

Purpose: TTitle can enable you to change the text in the title bar on the screen or window of the given port.

TWindow

Usage: TWindow -p <port> [-olc] [-w <width> -h <height> -x <X-pos> -y <Y-Pos> -f -s <size>]

- p the port to affect
- o to open the window or -c to close the window
- w the width of the window in pixels (optional)
- h the height of the window in pixels (optional)
- x the x-position in pixels of the top left corner of the window (optional)
- y the y-position in pixels of the top left corner of the window (optional)
- f the font to use in the window. This is case sensitive and must match the case of the font name as it appears in your FONTS: directory (optional)
- s the point size of the font to use (optional)

CLI/Batch file usable

Purpose: TWindow can open a window on the Workbench screen so that a port can be observed. This is functionally very similar to TScreen, except that TScreen opens a custom screen, while TWindow will open a window on your Workbench screen. No checking is done to make sure that the given width/height/position combination will fit on your Workbench screen. If you do not provide any of the optional parameters, DLG will use the configuration settings for the given port.

See also: TScreen

TWinHeight

Usage: TWinHeight -p <port> -h "<height>"

-p the port to affect

-h a string, enclosed in quotations, that specifies the height of the scrolling portion of the current window

CLI/Batch file usable

Purpose: This command is not meant to be used directly in a batch file or as a CLI command. It is used by ConfScr to modify the scrolling portion of the local window so that the local conference window does not obscure the main conference window. Note that the height parameter in this command is a string, and must be enclosed in quotes.

UNet2DLG

Usage: UNet2DLG

CLI/Batch file usable

Purpose: UNet2DLG handles the importing of UseNet newsgroups to DLG. Note that for every newsgroup you have active you must have a group name assigned (e.g. Comp.Sys.Amiga:) that points to the directory that the UUCP software uses for the newsgroup. When UNET2DLG is executed, it will go through your list of message areas and import any new messages in all of your newsgroup areas, deleting them from the source directory as it goes. UNET2DLG stores a file called ".dlg" in each newsgroup that indicates the number of the last message it imported. If you reset the pointer being used by your UUCP package (called ".next" in Matt's package) you must also reset the number in ".dlg" for that area.

UpdateAreas

Usage: UpdateAreas [-p -n -l -a]

-p forwards private NetMail to the user

-n forwards all NetMail to the user

-l invokes limited reporting. If a user is not a member of an area or has scrolled off the membership list for that area, they will not be informed of messages for them in that area.

-a forwards all NetMail regardless of its end FidoNet destination

CLI/Batch file usable

Purpose: This program should be run after a FidoNet session, unless you are using DLGMail. It makes DLG aware of new mail that has come in by setting message pointers, deleting excess messages, etc. It will also inform users of mail that has arrived for them on the system, and optionally moves it to their private mail areas.

Notes: Do not use CloseAreas in conjunction with UpdateAreas - the program does it's own opening and closing of message areas. You do not need to use this program in your FidoNet batch files if you are using DLGMail. DLGMail performs the same functions.

UserOptions

Usage: UserOptions

Menu item usable

Execute from menu as CHAIN

Purpose: UserOptions is the module on the default DLG main menu where users can modify aspects of their on-line accounts.

UUCP2DLG

Usage: UUCP2DLG

CLI/Batch file usable

Purpose: Scans the directory UUMAIL: and import any mail waiting there into the BBS. All mail will be imported to a user's private directory. If you are running UUCP on your system, all of your users have accounts of the form:

First_Last@path or alternatively ...path!First_Last

For example:

Joe_User@telepro.UUCP ...!alberta!herald!telepro!Joe_User

Also, groups are supported. For instance:

SysOp@telepro.UUCP

would send mail to all people in the SysOp group. It is recommended that you run this program after every UUCICO session.

ValSpeak

Usage: ValSpeak <infile> <outfile>

CLI/Batch file usable

Purpose: This utility translates an English message into a parody of "Valley Girl" slang. It is intended for fun only.

See also: Kraut, Jive

ViewArchive

Usage: ViewArchive <filename>

CLI/Batch file usable

Execute from menu as OVERLAY

Purpose: ViewArchive will view the contents of an archived file, provided that the archiver used was configured in the General configuration editor.

WaitPort

Usage: WaitPort -p <port> [-k <password>]

-p the port to wait for

-k the name of the password the port must have (optional)

CLI/Batch file usable

Purpose: Wait until an active lock on the specified port has the given password. If no password is given, wait until any lock is active on the specified port. WaitPort is breakable by a CTRL-C.

Who

Usage: Who

Menu item usable, Batch file usable in the context of an on-line session

Purpose: When run in the context of an on-line session, Who will display a list of users who are currently on-line, what port they are on, and what module they are using.

WMail

Usage: WMail [-b <threshold>]

- b indicate that a brief listing be shown if more than <threshold> messages are waiting. With a verbose listing, each message is shown as a three-line header. With a brief listing, each message only has one line.

Menu item usable, Batch file usable in the context of an on-line session

Purpose: When run in the context of an on-line session, WMail will show users what mail is waiting for them, and automatically tags the mail for tag reading. If you wish to run WMail automatically for users at log-in, you can put it into the DLGConfig:Batch/log-in.DLGbatch file as "DLG:WMail" on a single line. This will run it for each user as they log in. You might also elect to put WMail on your system as a menu entry. This would make WMail optional for your users. To install the WMail program into a menu, enter "DLG:WMail" as the string for the executable, and leave all other flags at their default values. A third option for using the WMail program would be to create a global command stack that would select the WMail menu item. This would make it operate for all users as they log in.

WriteLog

Usage: WriteLog [-p <port>] -c <code> -d <comment>

- p the port this entry belongs to (optional)
- c the log entry code
- d Comment for the entry. Maximum length of 35 characters

CLI/Batch file usable

Purpose: Writes custom entries into the userlog. This is useful if you want to log various TPTCron events into the same log that DLG uses.

XPRtransfer

Usage: XPRtransfer (slr) [-(fib) <filename>]

- (slr) s for send, r for receive
- f <filename> the name of a single file to transfer
- b <filename> the name of a file that contains a list of files to transfer.

sed internally by DLG

urpose: Facilitates file transfers using the popular XPR external protocol libraries. DLG will note what XPR protocol is being used, consult the custom protocol configuration files for the proper format, and transfer the file(s) in the appropriate fashion. For the most part, you will prefer to use the DLGZmodem module instead of the XPRZmodem library and XPRXmodem libraries, because it is faster.

Network Mail Overview:

DLG currently supports two different mail networks and this chapter will attempt to give you a general overview of both networks.

FidoNet:

FidoNet is a mail network organized and funded and maintained mainly by computer hobbyists. It currently has thousands of members world-wide and is very active and thriving.

There are two elements to the FidoNet network; private EMail (NetMail) and public message bases (EchoMail).

NetMail is very simply, a way for one user to send a private EMail message to another user of any other FidoNet node in the network. A message entered on your local system can travel directly to its destination (crashmail) by way of a direct telephone call to the receiving system, or indirectly (normal mail) through a series of organized routing methods.

EchoMail, on the other hand is a bit more difficult to explain. There are hundreds of FidoNet echo mail message areas. These areas would be similar in concept to a public message area on your BBS. They normally deal with a particular topic and all messages are available for reading by all users with access to the area. A FidoNet EchoMail area is a public message area that is mirrored on every system in the FidoNet network that subscribes to the area. For instance, if you entered a message in an EchoMail area on your local system, within days that message will have echoed to every other system in FidoNet carrying that particular area. In turn, all of the messages that are written on any system in the network carrying that area, will echo to your system. The end result is a very large global message area that is shared between hundreds and sometimes thousands of BBS's containing many thousands of users. There are several EchoMail areas whose focus is DLG. These provide a way for DLG SysOps to communicate and share ideas.

We are using the term "FidoNet" here in a generic sense. The original FidoNet network is called FidoNet, and uses protocols developed by Tom Jennings. Since the FidoNet has been created, several other networks have sprung up. These networks are separate from FidoNet, but they use the same software and protocols to achieve the same functionality that FidoNet has. These alternative networks usually have more of a focus for special interest groups. For example, during the Gulf War, a special SaudiNet was created to help people stay in communication with family members who were serving in the international forces. There is an AlterNet, an AmigaNet, EarthNet, and so on, each catering to various special interests that cannot or will not blend into the general rules and regulations of the larger and original FidoNet. DLG is quite capable of hooking into one or all of these networks using FidoNet compatible software. We refer to all such networks as FidoNet because they are all based on the same protocols and software. The rules and regulations of each of these different nets may vary widely, but from the software side, they are pretty much the same.

How To Get into FidoNet

The main difficulty here is that there is no set and given way of joining FidoNet, or any of the other FidoNet based networks. Since FidoNet is largely an ad hoc organization, there is no centralized body to which you can subscribe and from which to receive information on joining. FidoNet is broken up into zones, based roughly on geography, with each zone having an over-all co-ordinator. Each zone is broken up into a number of nets, based roughly on cities within the region, and each net consists of a number of nodes, which are the individual bulletin board systems within each net. In order to join FidoNet, you will need to contact the co-ordinator for your particular net and obtain the detailed information that you will need.

In order to join FidoNet, we suggest that you log into another bulletin board system in your area that already has joined FidoNet. Leave a message to that board's SysOp, asking for the telephone number of the area Net/Echo co-ordinator. This will get you a long way towards getting the information you need to join. If there is no BBS in your area that has FidoNet, then you will have to try calling a board in the closest city to you that does.

FidoNet Software

The software for FidoNet consists of several parts. This chapter will not go into detail about these parts, but instead, give a general overview of what they do.

The Nodelist:

The nodelist is a very large text file that is distributed as a part of the FidoNet network. It is the 'telephone book' of FidoNet and contains the network addresses of every system in the FidoNet network. DLG uses this 'telephone book' when dealing with several aspects of FidoNet.

Since the Nodelist is a very large text file, and a text file is a very inefficient means of data storage, it needs to be compiled down into something that is easier to search. This is accomplished by the use of a program called TrapList. TrapList is included on Disk3 of your DLG distribution and is detailed in the chapter about TrapDoor. Compiling your nodelist should be your first step in getting your FidoNet system up and running.

TrapDoor:

TrapDoor is the part of FidoNet that deals with telephone calls and the transfer of FidoNet network mail. It is the program that runs as a front door to your DLG system. It answers incoming calls from users and other FidoNet systems, and if need be, makes outgoing calls to other FidoNet systems. Trapdoor's entire purpose is to physically move FidoNet mail packets from one BBS system to another.

DLGMail:

DLGMail is the part of DLG's FidoNet software that deals with the packing and unpacking of the mail packets that FidoNet uses. DLGMail makes (packs) the mail packets that Trapdoor sends to other systems and unpacks (tosses) the mail packets that Trapdoor receives from other systems.

TrapList, TrapDoor, and DLGMail are the main three elements of FidoNet. Needless to say, there are a lot of details to be looked after when configuring your FidoNet system and therefore there are very large chapters dealing with these details following. Be sure to read and understand these chapters very carefully before attempting to configure your FidoNet system.

UseNet Network:

The UseNet network is a professional network that is largely driven by main-frame computer systems running on very expensive dedicated lines. UseNet consists mostly of universities and when looking for a UseNet feed, universities are quite often, a good place to start looking.

Like FidoNet, UseNet is divided into two sections. First there is UUCP Mail. This would be parallel in concept to NetMail described above in the section on FidoNet. The second area of UseNet is the NewsGroup. This would be parallel in concept to the FidoNet EchoMail area as explained in the FidoNet section.

DLG's interface into UseNet is somewhat simpler than that of FidoNet as most of the UseNet details are handled by a PD UseNet package ported to the Amiga by Matt Dillon.

This manual will not go into the details of setting up your UseNet site as it is covered in detail in the documentation that comes with Matt Dillon's UUCP package. The latest versions of this UUCP package are available on many BBS's as well as the Fred Fish PD disk collection.

Once the UUCP package is set up and running there are some very simple steps for interfacing it to DLG. The interfacing consists of the use of three programs; UUCP2DLG, and UNet2DLG and DLGUUSend. It is highly recommended to get the UUCP package fully operational independently of DLG before you attempt to interface the two.

UUCP2DLG:

This module is run from the CLI after you have completed an incoming or outgoing UUCP call. It takes no command line arguments and has no config file. Its purpose is to take the UUCP Mail (private mail) and send it to the users on the local BBS. Mail must be addressed to the underscored username on your BBS. For example, uucp mail sent to hank_reardon would be directed to the private directory of the user named Hank Reardon if he existed.

UNet2DLG:

This module is similar to UUCP2DLG except, that it works on NewsGroups. This command also takes no command line arguments and also has no config file. It is simply run in a batch file after an incoming or outgoing UUCP call. In order for UNet2DLG to function properly, you will need to have a DLG message area set up for each newsgroup and have made an AmigaDos assignment in your startup-sequence that assigns the name of each newsgroup to the UUCP directory that contains the news articles. A typical assign would look like;

```
Assign comp.sys.amiga.datacom: uunews:comp/sys/amiga/datacom
```

DLGUUSend:

This module you will never need to worry about as it is called directly from the DLG software. Its purpose is to prepare outgoing messages, both UUCP and Newsgroup articles for transmission. This will happen automatically and transparently as messages are written by users.

Incoming UUCP Calls:

Incoming calls to DLG are handled by a special user account. If a user account on your system is set to be a UUCP Client, it designates that username to be used for incoming UUCP network calls by a specific system. After the name and password are entered on this account the a batch file called 'DLGConfig:batch/UUCP.batch' is executed. The purpose of this is to transfer the call over the UUCP Getty program that will take over and do all of the UUCP session handling. After the completion of this batch file, the call is instantly terminated.

The second part of the book is devoted to a study of the role of the state in the development of the economy. The author argues that the state has a crucial role to play in the development of the economy, particularly in the case of developing countries.

The book is written in a clear and concise style, and is accessible to a wide range of readers. It is a valuable contribution to the literature on the role of the state in the economy, and is highly recommended for all those interested in the subject.

CHAPTER 1

The first chapter of the book is devoted to a study of the role of the state in the development of the economy. The author argues that the state has a crucial role to play in the development of the economy, particularly in the case of developing countries.

CHAPTER 2

The second chapter of the book is devoted to a study of the role of the state in the development of the economy. The author argues that the state has a crucial role to play in the development of the economy, particularly in the case of developing countries.

CHAPTER 3

The third chapter of the book is devoted to a study of the role of the state in the development of the economy. The author argues that the state has a crucial role to play in the development of the economy, particularly in the case of developing countries.

CHAPTER 4

The fourth chapter of the book is devoted to a study of the role of the state in the development of the economy. The author argues that the state has a crucial role to play in the development of the economy, particularly in the case of developing countries.

DLGMail

What is DLGMail?

DLGMail is a system of programs and a few simple scripts designed to handle all of the tasks required to maintain a full FidoNet node.

DLGMail, a system of distinct modules all controlled by a central process, isolates you from the configuration and script file nightmare associated with adding unrelated FidoNet utilities to most bulletin board systems. When used as the full system, DLGMail integrates the passing of configuration information and operation between various processes in such a way that operation is simple, straightforward and easy.

DLGMail, from its inception, was conceived as a mail processing system written specifically for the DLG Pro BB/OS. Unlike alternatives which require hacks and kludges to work with DLG properly, DLGMail works directly with DLG itself, as an integral part of the bulletin board operating system. DLGMail works silently in the background, behind the bulletin board system, and is unobtrusive to your users and computer system itself.

Structure

DLGMail is comprised of several executables:

DLGMail

This is the central controlling background process (and self-launching set of coprocesses) which maintains the operation of the rest of the DLGMail system. It runs the other modules listed below, and directs TrapDoor to make calls to other systems as necessary through its built-in call scheduler.

This executable features an ARexx port capable of responding to ARexx-generated commands, RX scripts and other programs. Commands received through the port are divided into action categories, and dispatched appropriately through the use of four separate queues and flags.

DLGMail has been written with AmigaDOS 2.0 features in mind. While it is perfectly acceptable to run it under 1.3, some features may not work, or work less conveniently than under 2.0.

DMC

This is your interface into DLGMail, the program by which you tell DLGMail what you wish it to do. DMC is nothing more than a short program that talks to DLGMail's ARexx port. In turn, DMC can report the status of the commands, whether understood and accepted, and for some commands, return an action status.

DMC can also display the built-in help provided in DLGMail.

DLGImp

This is the message importing module which takes raw Fido mail packets and bundles and turns them into messages readable on your DLG system. It can link the messages into reply chains if desired.

Unlike other mail processors which also export to other nodes, DLGImp will not perform this extra task - all incoming messages are simply tossed by DLGImp, and exporting is handled below.

DLGExp

This is the echomail message exporting module which takes the messages on your system and turns them back into mail packets for other Fido systems.

DLGExp is capable of handling passthru areas, multiple domains, and inter-network/inter-zone seenby stripping.

DLGNet

This is the netmail only message exporting module which takes the messages in your netmail directory and turns them into packets, file requests and file attaches for other Fido systems.

DLGNet is AKA-aware and will intelligently handle netmail destined to all networks you may be a member of.

DLGBundle

This module examines the outbound directory looking for *.PKT files (and which have address information stored in the filenote) and processes them according to the bundler control file. Packets and bundles can be routed, and the bundles built on a destination system- by-system basis utilizing ARC, ZOO, ZIP, LZH and LHA compression methods.

Major Non-DLGMail programs used in connection with DLGMail

DLGMail will only function when used in conjunction with DLG Pro BB/OS. It will not function by itself, or with other bulletin board systems.

TrapDoor

This program, included in the DLG package, is the front-end of your DLG BBS system (also called a mailer) and answers the phone on your designated FidoNet phone line. Through an intricate handshaking scheme, users can break into the BBS or mail can be exchanged, all automatically. DLGMail directs TrapDoor when to place phone calls and to what nodes the calls should be made. TrapDoor v1.80 or higher is required.

TrapList

This accompanies TrapDoor and is used to process the FidoNet compatible nodelist distributed weekly.

Setting Up FidoNet

First things first, of course, and that is to have your DLG Pro BB/OS set up and functional. Do not attempt to add DLGMail until you are satisfied that your bulletin board system is configured to your liking, and is functioning properly.

Step-by-step starter instructions are included later in the manual. For now, an overview will be given:

Creating Areas In DLG

You need to set up a minimum of three areas in DLG to be used in a typical configuration: One NETMAIL area, one BAD MESSAGES area, and one (or more) echomail areas.

In general, configure the areas like so:

NETMAIL:

50-100 messages, renumber at 10,000, set netmail area flag on. Because of the speed in which DLGNet conducts its affairs, there is really no limitation in size of your netmail area (except, of course, for hard drive space considerations).

BAD MESSAGES:

50-100 messages, renumber at 10000, no netmail or echomail flag.

ECHO AREA(s):

Capacity according to how busy the area is, renumber at 10000, set the echomail flag on, set to strip seen-by's, and do not set the netmail, UUCP or alias area flags. You may, if you wish, create custom origin lines for each area, though there isn't any need to do so for areas in your primary network.

You will need to set up custom origin lines for all non-primary network echomail areas, if for no other reason than to override DLG's insertion of your primary address in the origin string itself. Be sure and include the address at the end of the custom origin line in the proper format (Zone:Net/Node.Point).

Please bear in mind that while DLGMail "speaks Fido" that there are several Fido-compatible networks to belong to, and as such you'll have to make a decision early on as to what network will be your primary one where you'll be getting most of your mail (if you desire to connect to more than one) and configure DLG for that particular network in the SysOp Network Configuration editor. Your best bet, if you intend to join FidoNet at all, is to make FidoNet your primary network.

DLGMail Installation

The remainder of this section is provided as an overview of what the installation text has you do, and also how to edit the various control files to operate DLGMail.

Paths and assignments used with DLGMail

DLGMail uses assignments to reach all of the files and data that it manipulates, thus the following assigns must be present for DLGMail to work properly:

FIDO:

Where DLGMail expects all Fido-related executables (with the exception of Trap* programs.)

DLGMail, DMC, DLGImp, DLGExp, DLGNet, and DLGBundle executable are all located here.

The DLGMail.??? configuration files are stored here as well, including DLGMail.CFG, DLGMail.ARE, DLGMail.BUN, DLGMail.MRT, DLGMail.MAC and DLGMail.PNT.

Your KEY_DLGMail will also be present in this directory. Note that DLGMail will NOT operate without this keyfile.

FIDO:Trap/

This is a subdirectory within FIDO: and where all the Trap programs are placed during the installation process.

TrapDoor, TrapList, TrapDoor.KEY, ListAcct, ClearAcct, GetNode, SetPasswd, ListPasswd, SetConfig and EnumNodes are all located here

FIDO:Batch/

This is a subdirectory where all DLGMail executables expect to find any DLGMail related batch files and is created within FIDO: by the installation program.

NLDiff.DMB, NLNew.DMB, NLRecompile.DMB as well as all other optional *.DMB scripts (explained later) are located here.

MAIL:

This is the traditional FidoNet "master" assignment. The only pair of configuration files which are located here (related to DLG and DLGMail) are TrapDoor.CFG and TrapList.CFG.

MAIL:Outbound/

This is created by the install script and must be assigned to OUTBOUND:. This is where all outbound mail will be built.

MAIL:Inbound/

This is created by the install script and is where TrapDoor will put all incoming network mail and where DLGMail will look to process it.

LOGS:

This is the path where DLGMail and TrapDoor will create and maintain their logging file(s). This path is also utilized by other DLGMail utilities.

Report files generated by DLGMail will also be stored here. *.RPT files differ from *.LOG files in that logs are constantly added to, while reports are generated new each time they are written, and always reflect the latest configuration or operational information.

T:

This is the standard AmigaDOS temporary storage area, and the usual place most archivers build temporary files. It is meant to be cleared upon a reboot, so assigning this to a recoverable RAM disk or hard drive partition is NOT recommended.

ENV:

This is the standard AmigaDOS "environment variable" directory assignment. DLGMail maintains several environment variables for its own use (and your use, if desired) in this area.

WARNING: The ENV: assignment should always be made to a non-permanent (i.e. RAM:) device so that, upon a reboot, no left-over information will be restored

MSG:

This is DLG's assignment and is used to access the message bases on your DLG BBS.

PASS:

This is similar to DLG's MSG: assignment. This is where you'll place all of the areas your system will "pass through" without regard to the BBS. This is useful if you become a hub and move echos for other people and don't actually want to commit BBS areas and space to them.

Inside, you'll create subdirectories to handle the various passthrough'ed areas, and there is no usual limitation on the names. All of the following are 100% valid:

PASS:1
PASS:001
PASS:GENERAL
PASS:DLG_BETA
PASS:AMIGA

USER:

This is DLG's assignment and may be used to access the SysOp's own event log. DLGImp uses this to log incoming mail events, attached file moves, etc.

C:

This is where archivers such as ARC, PKXArc, ZOO, LZ, UnZip, etc. are usually expected to be found. It is not necessary to have these archivers in C:, provided you supply your own archiver configuration(s) including the path to the archivers.

Another speed tradeoff designed into DLGExp for sake of data security is to keep unused files closed and open them only when reading or writing. While this does tend to slow the processing of mail somewhat, the data is usually secure in case of a system failure and DLGMail can pick right up where it left off previously when restarted.

The FIDO:DLGMail.CFG file

All of the global configuration information, shared by DLGMail and its subordinate executables, is stored here. The file is opened and read ONCE, when the DLGMail executable first begins to run, and is not reread again unless you command DLGMail to "reconfig" for you, therefore you must remember that what you have configured and what DLGMail might be using, at any given moment, might not be the same thing.

To help you out, DLGMail produces two reports whenever it configures, a LOGS:DLGMailCFG.RPT showing you the settings you have configured (as well as all the assumed settings for items you didn't explicitly specify) and a LOGS:DLGMailCFGErr.RPT showing you any missing mandatory entries as well as keywords it doesn't understand.

General Settings

PUBSCN_NAME An AmigaDOS 2.0 feature which will allow DLGMail to open its status window and processing window on a previously opened public screen, or create one if the one specified does not exist. Omit this entry if you want these windows to open on the Workbench.

Example:

```
PUBSCN_NAME MyPublicScn
```

PUBSCN_LACE If opening a new public screen, specifies whether or not the screen will open in interlace mode. Default is 1, for an interlace screen.

Example:

```
PUBSCN_LACE 1
```

PUBSCN_OSCAN If opening a new public screen, specifies whether or not the maximum overscan as specified by preferences will be used. Default is 0, which specifies a screen size of 640 wide by either 200 or 400 (depending on the lace setting).

Example:

```
PUBSCN_OSCAN 1
```

PUBSCN_BEHIND If opening a new public screen, specified whether or not the screen will open in front of or behind all other screens. Default is 0, in front of, and 1 causes the screen to open behind all others.

Example:

```
PUBSCN_BEHIND 1
```

STATUS_X The upper left hand X coordinate of the status window. Unlike the old DLGMail, this window never resizes so it goes where you tell it and stays put. Default is 0.

Example:

```
STATUS_X 300
```

STATUS_Y The upper left hand Y coordinate of the status window. Default is 15.

Example:

```
STATUS_Y 94
```


STATUS_WIDTH How wide the status window will open to initially when DLGMail first loads. The Width can be adjusted later using the resizing gadgets. Default is 640.

Example:

```
STATUS_WIDTH 340
```

PLEASE NOTE: *The status window will not open if the dimensions you set for it exceed your screen size. STATUS_X + STATUS_WIDTH must be less than or equal to your screen size. The status window height is fixed at 106 pixels, and this value should be taken into account when selecting a value for STATUS_Y. STATUS_Y + 106 must be less than or equal to your screen height.*

PROCESSWINDOW This setting determines whether or not a separate mail processing window will open when mail is imported, exported, etc. Default is 1, which enables the window.

Example:

```
PROCESSWINDOW 1
```

PROCESS_X The upper left hand X coordinate of where the process window will open (if used). Default is 0.

Example:

```
PROCESS_X 310
```

PROCESS_Y The upper left hand Y coordinate of where the process window will open (if used). Default is 100.

Example:

```
PROCESS_Y 140
```

PROCESS_WIDTH The width of the processing window. Default is 640.

Example:

```
PROCESS_WIDTH 340
```

PROCESS_HEIGHT The height of the processing window. Default is 100.

Example:

```
PROCESS_HEIGHT 50
```

PLEASE NOTE: The processing window opens when necessary, and closes when not in use. The setting of PROCESSWINDOW itself kills the output from the subexecutables when the window is not open. Settings below will affect turning the subexecutable output off ONLY when the processing window is open.

ALSO PLEASE NOTE: *The process window will not open if the dimensions you set for it exceed your screen size. PROCESS_X + PROCESS_WIDTH must be less than or equal to your screen size, and PROCESS_Y + PROCESS_HEIGHT must be less than or equal to your screen height.*

ADDRESS Your main network address, in 4D format. No default can be specified for this, and this entry is mandatory, therefore DLGMail will not run unless you specify this. For new folks setting up who do not have an address at all, specify an address similar to <your zone>:<your local net>/9999.0.

Example:

```
ADDRESS 1:114/52.0
```

OPERATOR Specifies your name to the DLGMail system so you can be notified in your event log and elsewhere of activity and errors. This will also allow netmail addressed to SYSOP to be moved to your private directory as well. This setting is mandatory, and DLGMail will not run unless you specify this.

Example:

```
OPERATOR John Doe
```

CALLTYPES Specifies, hour by hour beginning at midnight, how to place outbound calls by flavor. This is a mandatory setting, which must be provided, or DLGMail will not run. Please see the original documentation on the setting of calling types, and specifically, the configuration settings for CRASHMAIL, NORMALMAIL, DIRECTMAIL and DIRECTONLY.

Call types is used to determine what calls will be placed whenever DLGMail receives a CALL or DELAYCALL command, or the built-in AUTOCALL triggers a call to be placed.

The format for specifying call types has changed, and now consists of a string of characters exactly 24 bytes long, one character per hour. Characters which are meaningful to DLGMail are:

C Crashmail only

N Normal, crash and file requests

D Direct, normal, and crashmail

Z Directmail only

- Don't place any calls this hour

As an example, assume the following: I wish to place "N" calls between 11 PM and 2 AM, "D" calls during the 2 AM hour, more "N" calls until 8 AM, no calls until noon, and "C" calls from noon until 11 PM. Remember that the first character corresponds to midnight.

Example:

```
CALLTYPES NNDNNNNCCCCCCCCCCCCCN
```

POINTNET If you feed points and some of those points require to be fed using the "pointnet" or "2D" kludge, you must pick a pointnet address and enter it here. If all your points will communicate with your system using 4 dimensional addressing, this is not necessary.

You should set the zone to your main address' zone, and node and point numbers to 0.

PLEASE NOTE: You DO NOT enter the pointnet address as an AKA.

Example:

```
POINTNET 1:5252/0.0
```

AKA You may belong to several networks, or because you are a hub or network coordinator, need to be recognized by several addresses. Configure DLGMail with up to 8 such addresses here (or 9 if you aren't using the POINTNET kludge above).

PLEASE NOTE: Do NOT repeat the pointnet address here!

Example:

```
AKA 1:114/1400.0
```

```
AKA 27:3010/2.0
```

ORIGIN The default origin string DLGMail will use if it ever exports a message in an echo area which is missing this line. You typically only specify your system name here, and DLGMail will prepend it with the " * Origin: " and your address.

Example:

```
ORIGIN Steve's One Stop DLG Shop
```

TDPOR The name of TrapDoor's ARexx port. Default is "TrapDoor" and is, of course, dependent on how you've configured TrapDoor.

VERY IMPORTANT - *The port name case is very important! "TRAPDOOR" is not the same as "TrapDoor" is not the same as "Trapdoor" and so forth. Make sure you set this correctly, and identically to that configured in the MAIL:TrapDoor.CFG file, or DLGMail won't be able to call out.*

Example:

```
TDPOR TrapDoor
```

BBSPOR The name of the BBS port which has TrapDoor associated with it, usually TR0 but may be different. Default is "TR0."

Example:

```
BBSPOR TR3
```

NETMAILDIR The message area number of the netmail directory, as configured for DLG. You must specify this setting as there is no default and DLGMail will not run without it.

Example:

```
NETMAILDIR 52
```

BADMSGSDIR The DLG area number where you'd like bad messages tossed to. This area should be set up as a local area only, and should be read-only as well. DLGMail requires this setting, and there is no default.

Example:

```
BADMSGSDIR 114
```

TASKPRI The Exec system priority value for all DLGMail, PDQMail and subordinate processes to run at - this specifies how much processor time DLGMail (and related) will receive relative to other tasks and processes on your system.

You want to hide DLGMail in the background, therefore want to enter a number here less than that of all common tasks and processes.

Under different versions of the operating system, this value may need to be changed. Default is -50. Useful values range from -1 all the way down to -127. Do not set this value greater than 5 (which would be an unwise setting to begin with). Default is a setting of -50.

Example:

```
TASKPRI -10
```

STACKSIZE Determines how much stack to allocate for DLGMail's child processes, the minimum being 12000 bytes. Unless you have DLGMail run a stack-hungry executable with a batch file, it should not be necessary to set this value any higher. Default is 12000.

Example:

```
STACKSIZE 15000
```

AUTOCALL How often (in seconds) DLGMail will attempt to make outbound calls. This may not be disabled, so remove all TPTCron cron table references to CALL unless you want to supplement this setting with some specific call times.

The useful range of this setting is between 100 and 1000 seconds, and your setting will be adjusted to fit this range. Default is 600.

Example:

```
AUTOCALL 600
```

AUTOCALLZONE What particular zone (if any) you want autocall dial attempts made to (default is 0, which disables limiting and DLGMail will then try to call into any zone for which mail is waiting when appropriate by time of day).

To limit calling to zone 1 only, use something like this:

Example:

```
AUTOCALLZONE 1
```

Please note: *If mail is packed up to be sent to other zones, you'll need to provide a manual or automatic way of causing calls to be placed to those zones. Appropriate use of the DLGMail commands DIRECT, NORMAL, etc. in conjunction with the zone argument, and caused to run during your time of day that other zones would be experiencing zone mail hour would be appropriate.*

TDSPAWN Indicates to DLGMail that you are running TrapDoor in BBSSPAWN mode, instead of BBSEXIT mode (See your DLG and/or TrapDoor documents). Default is 1.

Example:

```
TDSPAWN 1
```

POINTENABLE Enables echomail to be sent to points. Default is 1. Setting this to 0 effectively removes all the point entries in your DLGMail.ARE file without actually removing anything.

Example:

```
POINTENABLE 1
```

NOTIFYACTIVEONLY Determines how echomail notification will be handled. Default is 1, and causes only active users in specific echomail areas to be notified of arrival of echomail addressed to them. If set to 0, all users will be notified of all new echomail received in all areas, regardless of their activity in the various echomail message areas.

Example:

```
NOTIFYACTIVEONLY 1
```

VERBOSITY Determines the verbosity of logging and screen information displayed. Range is 0 to 5, default is 1.

Example:

```
VERBOSITY 2
```

QUICKLOG Buffers log writes in DLGImp and DLGExp to a few thousand bytes, saving time when a lot of logging is being done since the log file is only written to once in awhile instead of frequently. Default is 0.

If you have problems during runs of DLGImp or DLGExp, set this value to 0 so that the logging will always accurately reflect the last activity of these executables, thus enabling you to track down the cause of the problem.

Example:

```
QUICKLOG 1
```

PKTBUFFERLENGTH Determines the minimum size of the packet buffer used when importing mail. The buffer will grow only when necessary if this setting is low, and the buffer will be refilled frequently. Depending on your machine and hard drive speed, a slight improvement in performance can be realized by increasing this size. Useful size ranges from 10,000 bytes to 100,000 bytes. Default is 10000.

Example

PKTBUFFERLENGTH 50000

LINKREPLIES Enables the linking of echomail replies. This is the master setting, overriding any area-by-area settings in the DLGMail.ARE file. Default is 1.

Example:

LINKREPLIES 1

PKTRROUTE Determines the handling of mail packets which aren't addressed to any of your system's addresses (main or AKA). A setting of 0 causes these misaddressed packets to be deleted. A setting of 1 causes these misaddressed packets to be moved to outbound, where they will be bundled according to default or configured DLGMail.BUN bundle instructions, and sent out. A setting of 2 causes all packets to be tossed, regardless of the destination address within the packet. Default is 2. Note: Set this to 2 when you apply for a node number so that your net coordinator's message will be tossed on your system. Afterward, set this as you deem appropriate.

Example:

PKTRROUTE 1

NETMAILROUTE Determines the handling of netmail which is sent to your system but not addressed to your system. A setting of 0 causes this netmail to not be exported. A setting of 1 causes this netmail to be routed according to any applicable settings in your DLGMail.MRT or DLGMail.BUN files. Default is 0.

Example:

NETMAILROUTE 1

SAVEDAMAGED Causes incoming packets which are found to be corrupt to be saved for later examination to find the cause. Default is 0, no saving.

Example:

SAVEDAMAGED 1

LOGLINES When using DLGMail's TRIMLOGS command, sets the approximate maximum number of 80 column lines to remain in each log file (the number of lines is converted to bytes by multiplying this setting by 80). The newest lines are retained, the oldest purged. Default is 500 lines (40,000 bytes).

Example:

LOGLINES 50

ACTIVITYLOG Enables the building of ACTIVITY.DAT files inside each echo area for PDQTrack to analyze. Default is off.

Example:

ACTIVITYLOG 1

AREAFIX *(This command will only be relevant if you are running the PDQ Utility package available from Intuitive Software)* Enables/disables PDQAreafix. Default is 0, disabling PDQAreafix.

Example:

AREAFIX 1

TICK Enables/disables PDQTick (or the execution of "FIDO:Batch/GenericTick.DMB" if PDQTick isn't present). Default is 0, no tick program.

Example:

TICK 1

HUB This command will only be relevant if you are running the PDQ Utility package available from Intuitive Software) Enables/disables running PDQHub. Default is 0, no PDQHub.

Example: PDQHUB 1

NOCALLBUNDLE When set to 1, inhibits DLGMail from placing calls when DLGBundle is in operation. Default is 0, to allow calls to be placed while the bundler is operating.

SECUREIMP Default is 1, and causes the origination address of all echomail messages to be compared to those addresses in the distribution lists of your DLGMail.ARE file prior to tossing. If an address listed in an echomail message is not found for that area, the message will be tossed to your BAD message area. This security feature is disabled if set to 0.

Debugging Settings

DB_ALL Sets the debugging level globally for all DLGMail and PDQMail programs. Default is 0, no debugging output.

DB_DLGMAIL
DB_IMP
DB_EXP
DB_NET
DB_BUNDLE
DB_TICK
DB_AREAFIX
DB_HUB

These settings override the DB_ALL setting for individual executables. Default is the DB_ALL setting.

To disable debugging info in all executables except DLGImp:

Example:

DB_ALL 0
DB_IMP 1

Log File Enable Settings

LOG_ALL Enables logging globally for all executables. Default is 1, logging enabled.

LOG_DLGMAIL
LOG_IMP
LOG_EXP
LOG_NET
LOG_BUNDLE
LOG_TICK
LOG_AREAFIX
LOG_HUB

These settings override the LOG_ALL setting for individual executables. Default is the LOG_ALL setting.

To enable logging info in all executables except DLGImp:

Example:

LOG_ALL 1
LOG_IMP 0

Stealth Settings

SHOW_DLGMAIL
SHOW_IMP
SHOW_EXP
SHOW_NET
SHOW_BUNDLE
SHOW_TICK
SHOW_AREAFIX
SHOW_HUB

These settings override the PROCESSWINDOW setting for individual executables, turning off the activity output for the designated executable, but only when PROCESSWINDOW is enabled with 1. Default is the PROCESSWINDOW setting. You cannot enable SHOW_* when there is no processing window, thus these settings are only useful to turn output off when it would otherwise show up in the processing window.

Example:

```
SHOW_BUNDLE 0
```

Archiver Settings

It is hoped that you don't have to tinker with the archiver configurations, but if you do, this is how you'll need to do it.

{archiver}ADD Defines the calling string for use when adding mail packets to bundles when using this archiver.

The following are the possible archiver add definitions:

ARCADD
ZOOADD
LZHADD
LHAADD
ZIPADD
ARJADD

{archiver}EXTRACT Defines the calling string for use when extracting mail packets from bundles when using this archiver.

The following are the possible archiver extract definitions:

ARCEXTRACT
ZOOEXTRACT
LZHEXTRACT
LHAEXTRACT
ZIPEXTRACT
ARJEXTRACT

NOTE: DLGMail comes configured with suitable strings for all add and extract functions. To see the exact strings in default, please examine the DLGMailConfig.RPT report file generated by DLGMail each time it configures.

When configuring archivers, please observe the syntax required by the archiver, INCLUDING THE PATH TO THE ARCHIVER, and for extract definitions, include OUTBOUND: for the path to the destination and source files. Please also remember that most archivers are sensitive to case when setting their options, so take care in insuring that options are configured with the proper case. Please observe any specific requirements and limitations as listed for archivers in readme or addendum files as supplied with your version of DLGMail.

Example:

```
LZHADD c:lharc a OUTBOUND:%s OUTBOUND:%s
```

For those of you running AmigaDOS v2.04 and who have an aversion to placing 3rd party commands and programs into your C: directory, provide an "add" assignment in a startup-sequence script like so:

```
ASSIGN C: device:directory ADD
```

(device:directory is where you have your archivers)

For 1.3 users, you'll either have to place the archivers into C: to use the defaults provided, or edit the DLGMail defaults to replace C: with the actual path to your archivers.

Default Archiver Configurations

```
ARCADD c:ARC M OUTBOUND:%s OUTBOUND:%s
ARCEXTRACT c:PKXARC -r %s
ZOOADD c:ZOO aq:M OUTBOUND:%s OUTBOUND:%s
ZOOEXTRACT c:ZOO xSO %s
LHAADD c:LHA -2 -N m OUTBOUND:%s OUTBOUND:%s
LHAEXTRACT c:LHA -m x %s
LZHADD c:LHA -O -N m OUTBOUND:%s OUTBOUND:%s
LZHEXTRACT c:LHA -m x %s
ZIPADD c:ZIP -k -s -q OUTBOUND:%s OUTBOUND:%s
ZIPEXTRACT c:UNZIP %s
ARJADD [not configured as there is no ARJ creation utility as of this writing]
ARJEXTRACT c:UNARJ e -y %s
```

"Approved" Archivers and Versions

The following archivers have been tested and have been judged for use with this version of DLGMail. Use of other archivers is done at your own risk - please test any deviations from the "approved" list before relying on them!

Format	Approved	Unsuitable	Not Tested
ARC (extract)	ARC 0.23 PKAX PKXARC	TARC	
ARC (add)	ARC 0.23	TARC	
ZOO (extract)	ZOO 2.00		Newer versions
ZOO (add)	ZOO 2.00		Newer versions
LZH (extract)	LHA 1.11 LZ 1.92		LHArc LHUnarc
LZH (add)	LHA 1.11 LZ 1.92		LHArc
LHA (extract)	(See LZH list)		
LHA (add)	(See LZH list)		
ZIP (extract)	UnZip 3.1	UnZip 4.1	
ZIP (add)	Zip 0.93 (-s mode ONLY)		
ARJ (extract)			UnArj
ARJ (add)	<< Unavailable >>		

There are several "automatic archive determining unarchivers" around (PolyXArc, UnDo, others) which peek inside an archive and call the correct archiver based on signatures within the archive. If any of these has been renamed to a real archiver name, and the program calls the archiver using the AmigaDOS Execute() function, your machine will most certainly crash. Please insure that all archivers are actual archivers, and not one of these "intelligent" dearchiving programs in disguise.

Using the DLGMail Executable

You communicate with DLGMail through its command port, usually with DMC. DLGMail is ARexx-compatible, providing some unusual flexibility and the ability to communicate with it and query it from ARexx scripts and other programs.

The DLGMail ARexx port name is hard-coded as "DLGM_ARexx" (and the name IS case sensitive!). In most cases, it will be necessary to parse the string returned and ignore the numeric return code, almost always 0.

The Built-In "Instant" ARexx Commands

("Instant" commands, where no co-process is launched to process them)

Command	Pri	Template
QUIT	0	

Inhibits the creation of new co-processes, returns an error when additional commands are received, and when all co-processes have completed will cause DLGMail to unload and free all resources.

TRAPDUMP	0	<ON OFF>
----------	---	------------

This command waits until it can seize the DLG port which has TrapDoor, then does so. This command will tie up the DLGMail ARexx port indefinitely if a user is on line, therefore can bring your system to a grinding halt until the user logs off (or TrapDoor regains the port).

RECONFIG	0	
----------	---	--

Causes the DLGMail.CFG file to be re-read and used. Note that screen/window alterations will not take effect until DLGMail is unloaded/reloaded again.

WHEREIS	0	
---------	---	--

Used internally by DLGMail/PDQMail executables. The global configuration memory address is returned encrypted in the error code.

PUBSHELL	0	
----------	---	--

Under AmigaDOS 2.0+, if a public shell is in use, will cause a CLI window to be opened on it.

STATUS	0	<CPx TRAPDOOR>
--------	---	------------------

Queries the status of co-processes and TrapDoor. Specify the co- process you are interested in using CP0, CP1, CP2 or CP3, or "TrapDoor" if you need to find out if TrapDoor is loaded.

HELP	0	<command>
------	---	-----------

Displays help to DMC's shell on the usage and actions of the command you have specified. THIS IS THE MOST UP-TO-DATE WAY TO FIND OUT THE USAGE OF YOUR VERSION OF DLGMAIL'S COMMANDS, and the DLGMail manual should be relied upon only as a guide.

The Built-In "Immediate" ARexx Commands

("Immediate" commands, where co-process 0 (CP0) is launched)

Command	Pri	Template
REPORT	0	

Generates a LOGS:DLGMailConfig.RPT file.

ZMH	5	<ON OFF>
-----	---	------------

Enables/disables zone mail hour in TrapDoor.

MSGADDED	0	<TAGNAME AREA#>
----------	---	-------------------

Intended for DLG to call when a user writes an echomail message. The specified area is queued in a temporary file until the user logs off and the command USEREXPORT is sent.

The Built-In "Background" ARexx Commands

("Background" commands, where co-process 1 (CP1) is launched)

Command	Pri	Template
QUICKATTACH	0	<PATH:FILENAME> <FLAVOR> <ADDR> [.]

Attaches path:filename to addr using flavor specified. Multiple flavors/addresses may be used for each file.

Please note that if the PATH: has been specified as "OUTBOUND:" that the system now assumes your intent was to have the file deleted once sent. Thus, if you quickattach OUTBOUND:example.lzh to 555/1212, once 555/1212 successfully receives it, your file will be gone. Obviously, you cannot use OUTBOUND: as a path when files are to be sent to multiple nodes.

QUICKREQUEST	0	<ADDR> <FILE> [FILE.]
--------------	---	-----------------------

Generates a file request of file from addr.

The Built-In "Call" ARexx Commands

("Call" commands, where co-process 2 (CP2) is launched)

Command	Pri	Template
NODELIST	5	<DIFF NEW RECOMPILE>

Causes the nodelist to be processed using a FIDO:Batch/(method).DMB batch file.

CALL	0	[ADDRESS ADDR [ZONE <ZONENUMBER>]
------	---	-------------------------------------

Causes OUTBOUND: to be scanned for outbound files matching the configured calltypes for the hour, and calls placed. If the optional ZONE keyword and zone is specified, scanning and calling is limited to that zone only. If the optional address is specified, that address will be phoned, whether or not mail is present in OUTBOUND: or not.

DELAYCALL	0	[SECONDS] [ZONE <ZONENUMBER>]
-----------	---	-------------------------------

Similar to CALL. The optional seconds keyword specifies how many seconds to pause before actually commencing operations (default when not specified is 8 seconds). This command is intended to be called when a user logs off the BBS to cause calls to be placed.

CRASH	1	[ZONE <ZONENUMBER>]
DIRECT	0	[ZONE <ZONENUMBER>]
DIRECTONLY	0	[ZONE <ZONENUMBER>]
NORMAL	0	[ZONE <ZONENUMBER>]
REQUEST	-1	[ZONE <ZONENUMBER>]

Similar to CALL, but overrides the calltypes in effect at the hour when called.

TRAPDOOR	-5	<ON OFF RECONFIG ANSWER NOANSWER>
----------	----	---

ON enables, OFF disables TrapDoor. Since this command is imbedded within a co-process, this command will return immediately and it will be necessary to check the status of TrapDoor using the STATUS command above. Note that the priority is lower than all other commands in this co-process class, causing it to be acted on last should other commands be queued.

RECONFIG

will cause TrapDoor to re-read its configuration file (MAIL:TrapDoor.CFG).

ANSWER

sends TrapDoor a RINGS 1.

NOANSWER

sends TrapDoor a RINGS 9999 to cause it to ignore all but the most persistent callers.

ACCOUNTING

<CLEAR> [ADDRESS <4D address>]

This will clear TrapList's call accounting for you by running FIDO:Trap/ClearAcct with the appropriate arguments. It is necessary to specify CLEAR, else nothing will be changed. You may optionally specify an address (CLEAR ADDRESS address) to clear only a single address.

There is a script hook (see section A1.9.12) so that you may log the current accounting log if you desire before it is cleared.

The Built-In "Process" ARexx Commands

("Process" commands, where co-process 3 (CP3) is launched)

Command	Pri	Template
---------	-----	----------

NETSCAN	2	
---------	---	--

Causes the netmail directory to be scanned and new netmail exported.

PROCESS	1	
---------	---	--

Causes a "process cycle" to be performed. See documentation elsewhere on this cycle.

EXPORT	0	[[AREA# TAGNAME].]
--------	---	----------------------

By itself, causes all regular echomail areas to be scanned and new messages exported. Including area numbers or tagnames causes the scan to be limited to those areas only. Specifying an area number of 0 causes the passthru areas to be scanned (useful in case a crash were to have occurred during a previous process cycle).

The netmail area is also scanned in all cases.

CHANGEFLOW	-1	<4D_ADDR> <FLAVOR> TO <FLAVOR> [POLL [EXPORT]]
------------	----	--

Changes the flavor of flow files. See documentation elsewhere on changing flavors. Using the optional POLL argument, a flow file will be generated if none exist for the specified node, and all echomail and netmail areas will be scanned for new outgoing mail if the other optional EXPORT argument is provided.

CHANGEPKT	-1	<4D_ADDR> <FLAVOR> TO <FLAVOR>
-----------	----	--------------------------------

Changes the flavor of pkt files. See documentation elsewhere on changing flavors.

USEREXPORT	-1	
------------	----	--

Intended to be called from all BBS lines when the user logs off, this causes any echomail areas he has written in to be exported. The netmail area is scanned for export as well.

TRIMLOGS	-1	
----------	----	--

Reduces all LOGS:*.LOG files to approximately the number of lines defined in the DLGMail.CFG LOGLINES entry.

Environment Variables

The following environment variables are massaged while DMC is operating:

DLGMail

This environment variable is present whenever DLGMail is operating. It is removed when DLGMail closes.

GCFG

This environment variable contains data used internally by the DLGMail and PDQMail system. DO NOT ATTEMPT TO REMOVE OR CHANGE THIS DATA!

The DLGMail.ARE file

The DLGMail.ARE file is the heart of echomail distribution, and controls where an inbound echomail message will end up, and also where that message may be passed along to.

A typical DLGMail.ARE file looks like this:

```
MYADDRESS fido!1:114/52.0
tag AMIGA 12 114/77
tag PHX_SysOp 19 114/77
tag DLG_BETA 42 140/90 104/224
PASSTHRU
tag CLUB_AMIGA CLUB_AMIGA 153/910 280/322
```

Now for some explanation.

First, you need a line in your DLGMail.ARE file which contains your network, zone, net, node and point address, a sort of anchor address from which DLGMail can operate, thus the line:

```
MYADDRESS network!zone:net/node.point
```

Please keep in mind that the specification for "network" is local to the DLGMail.ARE file and can be virtually anything you wish. Be descriptive and it will assist you later. Do not include any spaces in the network name, nor the rest of the address. Be sure and include the point number.

MYADDRESS can appear any number of times within the DLGMail.ARE file - when the file is parsed, the last defined MYADDRESS is used until redefined. This is very useful when you belong to multiple networks. It must be defined at least once, at the top of the DLGMail.ARE file.

An echomail area entry in the DLGMail.ARE file (for a normal message area that BBS users can read) takes the form of

```
keyword TAGNAME MSG_AREA_# [modifier] address
```

An echomail area entry in the DLGMail.ARE file (for a PASSTHRU area) takes the form of

```
keyword TAGNAME SUBDIRECTORY_NAME [modifier] address
```

The keyword to indicate the beginning of a new area entry is either "tag" or "area" - you pick which you like best and use it consistently.

TAGNAME is the name the echo is distributed under.

MSG_AREA_# is the DLG area number used in the BBS.

SUBDIRECTORY_NAME is the actual subdirectory name created inside of your PASS: assignment to hold this particular area. DO NOT include the full path, nor add slashes or other symbols to the name.

The modifier is optional, and used to modify the behavior of the exporter when the area is processed. These will be explained a bit below.

The address is where you get/send the echo from/to. There can be multiple addresses (and multiple modifiers for each or several addresses). An address can take the form of something as simple as only a node or point number, all the way out to a full 5D address. The very last address the system parsed is used as a basis for "filling in" missing address components, thus:

If last address parsed was fido!1:114/52.0, and the next address is specified as 14, then the effective entry becomes fido!1:114/14.0

Similarly, if the last address parsed was fido!1:114/14.0 (above), and the next address is specified as 308/11, then the effective entry becomes fido!1:308/11.0

The following is a list of modifiers which may appear WITHIN the distribution list boundaries of any area.

CONT

Use this modifier at the beginning of a new physical line to indicate that you are continuing entries for the same area as in the previous physical line. This is useful to keep long distribution lists split over several lines, making editing easier.

```
USEADDRESS <domain!zone:net/node.point>
```

Use this modifier within the distribution list to indicate that the distribution addresses FOLLOWING this entry will receive the address specified within as the "from" address.

Assuming your regular address is mine!1:2/3.0, and a distribution list that looked something like this:

```
tag TAGNAME 313 114/77 USEADDRESS other!7:6/5.0 414/8
```

messages would be exported to 114/77 as being from 2/3, and messages exported to 414/8 as being from 6/5.

STRIP

NOSTRIP

Causes echomail to addressed between STRIP and NOSTRIP to have all previous SEEN-BY and PATH information removed from outbound packed mail. Only YOUR address will remain.

STRIP has little use except in cases of moving echomail between networks.

The default for each area is NOSTRIP. Note that it is not necessary to STRIP for other zones as mail passing between zones will be stripped of all SEEN-BY and PATH info.

Setting NOSTRIP will turn off the stripping feature for the remainder of the areas within your distribution list.

Nowthen, if that weren't enough, there are some global modifiers which appear BETWEEN area entries. These MAY NOT appear within an area entry:

```
MYADDRESS network!zone:net/node.point
```

Same as at the head of the DLGMail.ARE file, all remaining entries (unless changed again) are marked as having come from the newly defined MYADDRESS. Useful if you belong to more than one network, group areas from same networks together.

PASSTHRU

NOPASSTHRU

Causes messages being exported to be deleted, and the high water mark to be reset to zero for the area. Messages in areas appearing while PASSTHRU is active are NOT stored in MSG:area#/ like regular messages, rather are stored in PASS:name/ where PASS: is a new assignment and name is derived from the message area number entry in an alphanumeric fashion, such as:

```
PASSTHRU
tag 80XXX peecee_crap 114/77 114/8
```

DLGImp will, for PASSTHRU areas, use the tagname of the area for path if you substitute "[DEF]" for the area subdirectory name. Thus, the 80XXX echo could be defined like this:

```
tag 80XXX [DEF]
```

DLGImp will create PASSTHRU area directories (using the tagname of the area or as specified) if the directory does not exist. Please note that if the tagname of the message area contains colons or slashes, this method will NOT work, and so you'll have to specify a path.

The NOPASSTHRU setting counters the PASSTHRU setting, returning the remaining message areas back to normal.

LINK

NOLINK

Enables reply linking for all areas following, until a NOLINK setting is detected. Please note that linking must be enabled in the DLGMail.CFG file for linking to work. Also note that it is not necessary to initially set LINKing as LINK is the default.

A setting of NOLINK disables linking for all areas following.

CLASS class_number

This is used for PDQAreaFix (*available separately*) purposes - basically, allowing you to assign distinct numbers to groups of message areas, then giving areafix access to each subordinate node by specifying the CLASS numbers. Unlike most Areafix programs, the PDQAreaFix program will have a list of affectable class numbers associated with each node who can use it, so a node could be able to alter his settings of class 100 areas while NOT being able to add or subtract echos out of class 1 or 10 or 22.

Actual DLGMail.ARE In Use

During an import, the first two fields of information are used by DLGImp to determine where a message should be stored. If security checking is enabled, for every incoming echomail message processed, the distribution list is examined to determine if the node sending the message is somewhere in that list (and if not, the message is handled as a BAD piece of mail).

If a message comes in missing the AREA: field, it is tossed as netmail. If the AREA: field is present, the tag field of all areas in the DLGMail.ARE file is searched for a match. If the tagname is found in the DLGMail.ARE file, the message is stored to the path specified by the areanumber field. If the tagname is not found, the message will be tossed as bad.

During an export, each message is examined and its seenby list is compared to the distribution list (everything to the right of the area number) in DLGMail.ARE. Any nodes appearing in the distribution list which are not listed in the SEEN-BY list of the message will be exported.

DLGNet/DLGBundle Configuration and Control Files

DLGNet handles the processing of NETMAIL only, turning a netmail message into a packet which may be ready to send or need to be bundled into a group of packets being sent to a particular node.

DLGBundle takes all the packets created by DLGExp and some of the packets created by DLGNet and either renames them into a suitable transmittable packet name or combines them with none or some additional packets destined for some node by using one of the common archivers, turning them into bundles.

It's truly impossible to separate DLGNet and DLGBundle (from a configuration point of view) because both executables share a common control file, the DLGMail.BUN file.

Before we start discussing configuration, however, a more thorough discussion of DLGNet and DLGBundle is in order:

DLGNet

DLGNet's job is to process netmail, whether it originated on your system or was imported from another. Netmail, unlike echomail, constitutes a NODE TO NODE form of communication, rather than echomail's "broadcast" method. Processing is different in this respect, since for any given message we know where it came from and where it is headed.

Over the years, and in response to driving the cost of transmitting netmail across the street, nation or world downward, FidoNet has created and expanded methods by which a message from your system destined anywhere may be routed through other systems, presumably with already established paths between the two nodes in question, thus cutting the cost of moving a single message down from what could be a full minute of long distance connect time to what would amount to a fraction of a minute added to an existing transmission. Confused? Didn't mean to cause that.

To move echomail across the country there already exists a "backbone" of interconnected systems. If this backbone (or other similar interconnection structure) is utilized for netmail, then the cost of sending a netmail message approaches zero.

DLGNet provides for routing of netmail in two ways: The first and most common (and always safest) is to route an individual message by including that message in a packet already destined for another system willing to route netmail. When the other system receives the message, discovers it is netmail and not destined for it, it will pack the message up and send it on, presumably "upstream" on the way to its eventual destination. The message will likely be unpacked and repacked many times along its way, but will make it.

Some nets provide a formal structure for this routing (sometimes referred to as "low priority netmail") so check with your NEC and NC to confirm that it is possible to route.

The second, and somewhat more interesting means of routing netmail (and echomail!) involves packing a message within a packet destined for the final system, and including the packet with other packets destined for different systems. This method is, shall we say, limited and is also not widely available. It is important to discuss it, however, so you gain an understanding of what happens with the interaction of the two control files.

DLGBundle

DLGBundle's job is to take packets which, at the point they are recognized by DLGBundle, "orphaned," and to complete the bundling and packet routing which may be called for. In addition, DLGBundle observes your wishes of how to bundle packets, and what transmission flavor you desire for them.

Messages, Packets, Bundles and Flow Files

Up to now you pretty much haven't needed to concern yourself with what constitutes a packet, bundle, message, flow file or whatever. To completely understand what DLGBundle does, some explanation is in order:

There are four major "categories" of files used in transmitting an idea from a user on one system to a user on another system:

A Message is stored in one of your message areas, such as MSG:5/15.MSG - this is the heart of transmitting ideas between people, and the same is true when you add Fido to your BBS.

An example would be as above, 15.MSG, which contains a single communication between two (or more) people.

A Packet consists of a number of messages. It is not efficient to transmit single messages around the network, therefore messages are combined into packets with a common destination. For instance, if you send three netmail messages to NODE G, two messages to NODE X and one

message to NODE Q, three packets would be created. One packet would contain the three messages to NODE G, another packet would contain the two messages to NODE X, and the other would contain the single message to NODE Q.

The concept to keep in mind here: A packet always contains messages meant to be tossed into the message base of a single destination system (NOTE: Some of those messages may be intended for systems further on, but they are to be sent to this particular system for processing FIRST. This is known as routing and is used for netmail ONLY).

Packets may contain only netmail, only echomail or a combination of the two - it does not matter.

Packet names will vary. All of these are legitimate packet names:

```
9f42e7b5.PKT
00720034.OUT (2D, old style, not used)
1.280.322.0.CUT
```

Packet names ALWAYS end in one of two extensions, either .PKT or .?UT where the ? is replaced by a letter signifying the "flavor" of the packet.

Packet names ending in the .PKT extension will be found inside of bundles (see below) or at the receiving end of a Fido transmission when an unbundled packet is transmitted.

Packet names ending in the .?UT extension are only found in the OUTBOUND: directory of the sending systems, and signify that the packet is ready to be sent and that no other processing is needed (or desired).

A Bundle contains one or more packets of messages, and are created for two reasons: First, so that mail accumulating for a particular destination system doesn't keep getting added onto a single packet which could soon grow to ridiculous size; second, by virtue that the bundle is created by using one of the common archival utilities such as ARC or LZ, the packets will be shrunk and require less storage room at both ends of the connection and less time to transmit (and that translates into cost savings for long distance connections).

Bundles almost always contain only packets for a single destination system, however DLGMail will allow you to include packets for more than one system in a single bundle (called Packet Routing). BEWARE! Not every mail processing system you will connect with will support such a scheme, therefore it is important you understand the distinction between routing netmail (on the message level inside a packet) and routing packets (on the packet level inside a bundle).

Bundle names look like so:

```
0000003D.SU5 (2D format, not created by DLGMail)
1.114.52.0.TH3
8.7301.313.0.M00
```

Notice that all the bundle names end in a special extension, .XXZ, where XX is the first two letters of the day of creation of the week (MO is Monday, TU is Tuesday, WE is Wednesday, etc.) and Z is the "bundle counter number" - If a fresh bundle is created on Tuesday, it will be named TU0. After that bundle is delivered, the next bundle created for that particular node on Tuesday will be named TU1, and so forth. The provision for the counter stems from the possibility that while the two systems may connect and exchange mail many times during a day, the destination system may not be able to process it immediately. This allows for 10 bundles to be received and remain untouched during any given day by a receiving system before the mailer has to resort to special naming conventions.

DLGMail uses the 4D naming technique when building bundles for destinations, however due to several factors we won't bother considering here, the bundle name must be converted to the old style 2D format for transmission - this is handled by TrapDoor automatically, and explains why you ONLY receive the 2D style bundle names when TrapDoor connects and is sent bundles from other systems.

Flow Files are nothing more than a special list of "things the mailer is to transmit" once a connection with the destination system is established. Because of the naming convention used, the flow file also establishes the manner in which a connection to a destination system is sought.

The following are legitimate flow file names:

```
1.114.52.0.HLO
0072003D.CLO (2D, not used with DLGMail)
```

Note that the flow file ends with the .?LO extension. the ? signifies the "flavor" of the transmission just as is done with packets.

A flow file is NEVER transmitted, rather it is used first by the outbound scanner to indicate mail is waiting for a destination, then secondly once a connection is made tells the mailer what files to send. As each file is transmitted, this file is altered and the entry for it removed.

It is extremely important that you understand the distinctions between bundles, packets and flow files - many operators do not and when a problem develops and it comes time to sort out what happened, it's difficult for two SysOps to communicate effectively when one uses the incorrect terminology. It also makes the offending operator sound unprofessional, green or, worst of all, ignorant of what his system is actually doing, and (unfortunately) many FidoNet SysOps are not tolerant of this sort of behavior.

Wildcards with DLGNet and DLGBundle

It is often desirable (and downright necessary) to be able to specify a number of nodes at a time for routing purposes. If routing were to require you to have a separate entry for every possible node in FidoNet, you'd need a routing file that was close to 10,000 lines long!

Through the clever use of wildcards you are already familiar with, it is not necessary to create such an unmanageable file at all. Observe:

1:*/.* Match all addresses in Zone 1.

1:114/.* Match all addresses in Zone 1, Net 114.

:114/. Match all addresses in Net 114 of any zone.

:/.* Not very practical, will match every address in the universe. There IS a use, however.

Further, the '?' may also be used, though of what use this would be has yet to be determined.

1:11?/.* Match any address in Zone 1 with a three digit NET number beginning with 11.

Thus, with careful construction of routing files, you can pretty much eliminate many entries with a carefully planned few. More on this later, however.

Routing takes place at the netmail-message level in the DLGMail.MRT file and at the packet level in the DLGMail.BUN file. To be explained later, the DLGMail.BUN file is scanned first whenever possible routing is to take place, so the use of wildcards in this file can be treacherous if adequate consideration is not given.

Also, please keep in mind that if you use wildcards and a netmail message destination matches a wildcard pattern in the .BUN file, routing will NOT take place in the .MRT file. Be careful!

The DLGMail.BUN Control File

This control file is used to control DLGBundle's bundling and packet routing features. IMPORTANT: It is used for netmail routing in the first pass of DLGNet to resolve whether or not DLGNet needs to make pass two through it's own control file. More on that a bit later.

The DLGMail.BUN control file has a simple layout:

```
<Address1> [THRU <Address2>] [ARCHIVER <extension or 'NONE'>] [TYPE <flavor>]
```

Note that Address1 MAY contain wildcards, while Address2 MAY NOT.

Very important: All addresses MUST be fully expressed in 4D format, as in 1:114/52.0 - if you forget the ".0" then results will be unpredictable at best.

Indeed, you need not even have everyone with whom you connect with in this file at all, as certain assumptions are made for you (though these are costly assumptions). First, the archiver will default to ARC, universally available on all platforms, and second, the flavor type will default to DIRECT, causing your system to phone the destination during zone mail hour to deliver the mail.

Since it is not desirable to be exchanging echomail during zone mail hour (and prohibited in many cases) you should have an entry and provisions for all nodes with whom you exchange echomail with to attempt to connect at times other than zone mail hour. Similarly, you can configure DLGMail to place outbound calls to nodes designated as DIRECT exclusively during zone mail hour using DIRECTONLY in the configuration file for zone mail hour. If you choose this method, make sure you flavor packets and bundles appropriately for those nodes which answer only during zone mail hour (while rare, there are still a few).

As a more practical matter, seldom are the defaults what you really wish to do for the folks you connect with on a regular basis. For instance, consider the following examples:

Case 1: Your echomail hub accepts crashmail and desires you to use a more efficient archiver, such as LZ to create LHArc-style mail. You'd want to use

```
1:114/77.0 ARCHIVER LZH TYPE CRASH
```

This will create an LZH bundle with accompanying flow file, and flavor the flow file for immediate (CRASH) delivery.

Case 2: You call the DLGMail support board once a day or less to pick up the DLG echos and DLGMail echo. To prevent calls during most of the day (to cut down on costs) and to only successfully complete a single call in a period (instead of possible several times) you'd want

```
1:114/52.0 ARCHIVER LZH TYPE HOLD
```

This will create an LZH bundle with accompanying flow file, and flavor the flow file for holding on the system. Until this flow file is altered, no attempt by your system will be made to call this node (See the CHANGEFLOW command below to find out how to deal with this).

Case 3: You swap mail with a local Macintosh board, and the operator doesn't have a reliable ARC utility he can use, nor does he have any of the other archivers for his platform. He also wants to exchange mail after 11 pm and before 8 am. You'd want

```
1:114/25.0 ARCHIVER NONE TYPE NORMAL
```

This will NOT bundle packets to him, rather create one giant packet flavored for normal (night) delivery.

Case 4: To circumvent needing a bunch of entries in the .MRT file, you'd like to be able to handle netmail for your local net in something of a global fashion, also setting up a default archiver and flavor type. Entries preceding this one override this, of course, so it's fairly safe though not foolproof:

```
1:114/*. ARCHIVER ARC TYPE CRASH
```

The major problem with an entry like this in the DLGMail.BUN control file is that it prevents you from, on an individual message-by-message basis, routing netmail to problem nodes which your TrapDoor won't connect with (and there ARE mailers out there, such as Star-Net, which refuse to cooperate with various versions of TrapDoor).

Note that in this case no "thru address" is specified, thus no routing takes place when packets match. An entry such as this simply supplies an overriding archiver and flavor for all entries which it matches.

Case 5: (WARNING: This shows you the use of packet routing, which as explained above is not terribly practical and can only be used under very specific circumstances. For simplicity, node numbers have been replaced by references to NODE A, B and C). You are NODE A (for this example) and wish to receive a busy echo from NODE C. You don't normally connect with NODE C, but DO connect with NODE B who connects with NODE C. Hmmmmmmmm. The usual way to get an echo is for NODE B to receive it, process it, and toss it to your node, but let's say the echo is of no interest to NODE B at all. If all three of these nodes used DLGMail, then it would be possible to route the echo in packet form through NODE B like so:

(NODE A entries:)

```
NODE_C THRU NODE_B ARCHIVER LZH TYPE HOLD
NODE_B  ARCHIVER LZH TYPE HOLD
```

(note that archiver entries and types must match)

(NODE B entries:)

```
NODE_A ARCHIVER LZH TYPE HOLD
NODE_C ARCHIVER ARC TYPE HOLD
```

(in this case, the archivers can be different)

(NODE C entries:)

```
NODE_A THRU NODE_B ARCHIVER LZH TYPE HOLD
NODE_B  ARCHIVER LZH TYPE HOLD
```

The case above illustrates some extreme versatility.

The DLGMail.MRT Routing File

Routing and wildcarding takes place identically as above, though this time we are not concerned with the flavor of packets or type of archiver, only where NETMAIL should go. Routing takes place on a message-by-message basis, and so long as the destination you wish to route mail through finds routing acceptable (especially for the addresses you intend to route), this is a safe and entirely acceptable way to reduce the cost of and speed up transmission of the lower priority netmail. NOTE that netmail created with the CRASH bit set WILL NEVER be routed!

The format for lines in the file is simply:

```
<ADDRESS> THRU <ADDRESS>
```

Consider this entry:

```
1:114/33.0 THRU 1:114/77.0
```

Netmail addressed to anyone on 1:114/33, unless you set the crash bit when entering the message, will be sent in a packet to 1:114/77 where 77 will then pack the message back up and send it on to 33.

Consider this entry:

```
1:114/*.*
```

Yes, that's the entire entry, and it is used for exceptions and causes DLGNet to treat the message as if it were routed, except that no routing takes place at all. You'll recall a similar entry in the .BUN file that looked like

```
1:114/*.* ARCHIVER ARC TYPE NORMAL
```

The effect is very similar, and DOES allow problem nodes to appear in the .MRT file ahead of it.

Consider this entry:

```
1:114/*.* THRU 1:114/77.0
```

now netmail addresses which fail in the .BUN file and fail in the .MRT file previous to this line will all be routed through 1:114/77.

Recall that we can expand on the wildcard idea fully, thus:

```
*:*/.* THRU 1:114/77.0
```

would simply send all netmail failing ALL previous tests to 1:114/77.

For a more practical application of what we're doing, we'll play pretend. Below are two fictitious routing files which could exist on anyone's system. For sake of argument, the system's address is 1:123/456.0.

DLGMail.BUN:

```
1:123/10.0 ARCHIVER ZOO TYPE CRASH ;his backbone feed
1:114/52.0 ARCHIVER LZH TYPE HOLD  ;his DLG feed
1:273/611.0 ARCHIVER LHA TYPE HOLD ;his "Amiga" feed
```

DLGMail.MRT:

```
1:114/*.* THRU 1:114/52.0      ;routes all netmail to net 114
                                ;in zone 1 through 1:114/52.0
1:273/*.* THRU 1:273/611.0      ;do same with net 273
;we know that 1:140/90, 1:153/910 and 1:280/322 all connect to
;node 1:114/52.0, so let's route any netmail to them there.
;further, we know that these nodes except 280/322 can further
;route, so let's be extra bold.
1:140/*.* THRU 1:114/52.0
1:153/*.* THRU 1:114/52.0
1:280/322.0 THRU 1:114/52.0
;Net 123 allows routing of overseas netmail through 123/5. We
;want to take advantage of that, but we must be careful because
;we sometimes want to send netmail to The Network which is in
;zone 8. The following will do this for us:
8:*/.* THRU 1:888/7.0           ;this guy will gate the mail
;into The Network for us
;nowthen, send all other netmail out through 123/5.
*:*/.* THRU 1:123/5.0
```

Keeping in mind that the very first time an address matches wildcards or a specific address in either the .BUN or .MRT file, that's the way the netmail will be routed. The .BUN file is evaluated first, followed by the .MRT file.

The following brief set of examples will show you the route that a netmail message would take, if netmail were sent using the control files from above.

```
1:123/10 -> 1:123/10 CRASH BUNDLE
2:860/5 -> NETMAIL TO 1:123/5 DIRECT BUNDLE
1:114/52 -> 1:114/52 HOLD BUNDLE
1:140/19 -> NETMAIL TO 1:114/52 HOLD BUNDLE
2:273/444 -> NETMAIL TO 1:123/5 DIRECT BUNDLE
1:273/15 -> NETMAIL TO 1:273/611 HOLD BUNDLE
1:123/47 -> NETMAIL TO 1:123/5 DIRECT BUNDLE
```

Use only what you need to get started and build on your routing scheme over a period of time.

VERY IMPORTANT! READ THIS CAREFULLY!

To configure the zone gating with DLGMail, you'll need entries in your DLGMail.MRT file, though if your net provides for the routing of netmail (sometimes called "low priority netmail") it won't be necessary, and maybe even counterproductive.

In every Fido zone there are "gateways" to other Fido zones - the idea being that these gateways connect daily and make it unnecessary to attempt overseas calls yourself. This method has advantages, but would appear to suffer from reliability problems from time to time.

In any case, to provide zone gating yourself you'll need to create entries in your DLGMail.MRT file (probably toward the bottom) according to the following formula:

X is your zone

Y is the destination zone

Y:*/*. * THRU X:X/Y

You'll need one of these entries for each "other" Fido Zone. For North America, the table of routing statements is as follows:

2:*/*. * THRU 1:1/2
3:*/*. * THRU 1:1/3
4:*/*. * THRU 1:1/4
5:*/*. * THRU 1:1/5
6:*/*. * THRU 1:1/6
7:*/*. * THRU 1:1/7

Using the formula and table above, DLGMail users in other zones may construct a similar table.

For the really ambitious, zone gates exist for The Network, RBBS-PC net, and others — it should be trivial to provide routing to those zone gates as well.

The DLGMail.PNT Routing File

The object of this file is to provide "point routing" of netmail to folks who feed their point systems off yours. If they receive netmail which is not fully addressed to their point, a copy of that message will be created and sent to their point anyway (if, of course, you have set them up as discussed below). The format of the file is quite simple:

Bill Williford TOPOINT 2
Benjamin Franklin TOPOINT 3
Alford Faffard TOPOINT 4
Stacey McIntosh TOPOINT 98

Please note that routing in this fashion causes DLGNet to assume that these points are 4D in nature. Regardless of properly point-addressed netmail and netmail created through this routing scheme, if you feed any points configured with 2D addresses, you'll need appropriate entries in the DLGMail.BUN and DLGMail.MRT files to cause creative routing to take place, notably not cause mail to the 4D address to be added to the 2D kludge packets. You therefore, won't want any entries for any points in the DLGMail.BUN file, but will want one entry each in the DLGMail.MRT file. See the example configuration files for assistance.

The Process Cycle

Whenever a successful connect is made to another mail node, TrapDoor causes a process cycle to be run. This is, as of TrapDoor 1.80, indiscriminate - whether any mail came in or not, the process cycle will run (a limitation of this version of TrapDoor).

It is important for you to know the order that mail is processed on your system when the DLGMail PROCESS command is sent.

First, INBOUND: is scanned for files. If no files are present, processing is discontinued.

The types of files present are examined. Mail files (packets and bundles) will set a DLGImp flag. *.TIC files will set the TICK flag. Other files will set the ATTACHED FILE flag.

Temporary files that TrapDoor creates and uses will not set any of these flags, and these files will be ignored.

If the DLGImp flag is set, DLGImp is called, and the packets and bundles are processed as appropriate. As mail is tossed, other flags may be set as well, including the DLGNet flag (if netmail is received) and DLGExp flag (if echomail is received). Echo areas are "remembered" so that DLGExp will process only the areas for which mail was received.

If you are using PDQAreafix, any netmail addressed to "Areafix" or "PDQAreafix" is remembered and a flag is set for PDQAreafix.

Mail is then linked.

If the PDQAreafix flag is set, PDQAreafix is run and processes messages received which are addressed to it. It builds reply messages which will be exported by DLGNet later. (*The PDQ Utilities package is a separate product produced by Intuitive Software*)

If the TICK flag is set, PDQTick will be run next. Since PDQTick can build netmail and echomail replies, flags for DLGNet and DLGExport will be set as appropriate.

Please note that PDQTick will run no more than twice, no matter how many times a process cycle repeats when launched.

If FIDO:PDQTick is not present, the "FIDO:Batch/GenericTick.DMB" script will be executed instead.

If the File Attach flag is set, PDQHub is run - and depending on how you have it configured, will move files around and create event log entries, message replies, etc.

Please note that PDQHub will run no more than once, no matter how many times a process cycle repeats when launched.

If the DLGExp flag is set, Echomail areas will be scanned and exported according to the "memory" of what had previously been imported or created.

If any packets are created, the DLGBundle flag is set.

If the DLGNet flag is set, your netmail area will be scanned and exported.

If any packets are created, the DLGBundle flag is set. If any netmail is scanned which is flagged as crash, the CRASH flag is set.

If the DLGBundle flag is set, DLGBundle is run, to cause the created packets to be turned into 4D packets and bundles, ready for sending.

If the CRASH flag is set, a new CALL co-process will be launched, and DLGMail will attempt delivery of the crash mail.

At any time during the process cycle, any executable *may* set the QUIT flag which will cause processing to abort. If you notice this happening, examine the logs to determine which executable is causing this, and the reason for it.

Nowthen, INBOUND: is reexamined to see if any new files have come in during the initial run of processing. The cycle repeats until INBOUND: has been cleared of all mail and attached files. If, for some reason, it is impossible to clear INBOUND: of these items, processing is discontinued after two complete cycles so your system won't loop endlessly.

The DMC CHANGE Commands

There are times when you need to change the flavor of outbound mail on your system, the most obvious case is when you build HOLD mail and need to send it off at some special time.

Within DLGMail are provisions to do this, accessed through DMC like other mail processing commands.

To change the flavor of packets (the files that end in "UT") is the command "CHANGEPKT" - it ONLY affects packets and leaves flow files alone.

To change the flavor of flow files (the files that end in "LO") is the command "CHANGEFLOW" - this only works on flow files and leaves packets alone.

The syntax of these commands is very similar

```
CHANGEPKT <address> <from> TO <to>
```

```
CHANGEFLOW <address> <from> TO <to> [POLL [EXPORT]]
```

Addresses used to change flavors of mail have the most wildcard flexibility of any addresses used in the DLGMail system.

First, think of a 4D address in terms of 4 individual components, the zone component, net component, node component and point component. Your main address zone component is implied, if no zone is specified, and your main address net component is implied if no net is specified. While not very useful, your own node number component is implied if no node is specified. A point component of 0 is implied if no point number is specified.

For maximum flexibility, two different wildcards are available for use:

*Matches any component

#?Matches any number of components including and after that specified

DLGMail assumes certain things based on the standard 4D address "punctuation" - if no punctuation surrounds a number, it is assumed that the number is a node number. Expanding from that will always be punctuation, so the intent will become clearer.

Given this much information, and assuming your own address were 1:555/1212.0, the following addresses (used in changing flow) would act in the following manner:

Address As Specified To Change	Actual Address(es) Matched
1000	1:555/1000 (Your zone and net are implied)
.5	1:555/1212.5 (Your zone, net and node are implied)
*	All nodes in zone 1, net 555 (Your zone and net are implied)
280/322	1:280/322
280/*	All nodes in zone 1, net 280 (Your zone is implied)
1:*/100	All nets in zone 1, node 100
1:#?	All nets and nodes and points in zone 1 (Same as 1:*/*.*)
2:310/#?	All nodes and point in zone 2, net 310

As you can see, this scheme is very flexible.

The <from> is a letter designating the flavor of packet or flow file you desire to search for (given the address). C for crash, N for normal, H for hold, D for direct and ? to match any flavor.

The <to> is a letter designating the flavor of the packet or flow file you desire to change a matched file to. For obvious reasons, the ? is not allowed.

In the case of CHANGEFLOW, you can optionally tack "POLL" to the end of the command. If no flow file is found, an empty flow file will be built (with the flavor of <to> as you specified) which, depending on time of day, can cause your system to poll another system for mail.

Also in the case of CHANGEFLOW, you can optionally tack an EXPORT onto the command *if* you are using POLL. This will cause all echomail and netmail areas to be exported. This is normally not necessary, but the option exists should you find a use for it.

While you can type the DMC command at the CLI prompt, you'll find its primary use is in your CronTab to cause your system to change flavors on specific days at specific hours. You may also consider "programming" these change commands into your DLGMail.MAC macro file, and using descriptive names in your CronTab as defined in the .MAC file. For instance, you might have the following in your .MAC file:

```
POLLMIKE      ChangeFlow 202/503 ? to D POLL
HOLDMIKE      ChangeFlow 202/503 ? to H
```

To cause the system to poll this fellow around 2 AM, and give up at 5 AM, you'd then have CronTab entries like so:

```
0 2 * * * Fido:DMC POLLMIKE
0 5 * * * Fido:DMC HOLDMIKE
```

The example above assumes that you have configured the bundler to build "hold" mail for Mike.

DMC and DLGMail Overview

DLGMail is a "resident" executable. Like TPTCron, DLGMail is launched in its own window when the BBS is started. Like CronEvent talks to TPTCron, DMC talks to DLGMail and DLGMail manages everything.

Invoking DLGMail

The s:FidoStart.Batch file will create a new CLI window and launch DLGMail for you. Please note the installation instructions for the new placement of the "execute s:fidostart.batch" line.

After some rudimentary checks on stacksize, etc., DLGMail will assume control over its window and proceed to configure itself.

Communicating With DLGMail

You control the FidoNet activities of your system using a program called DMC. DMC, in turn, tells DLGMail what it is to do.

From CLI:

```
1> DMC COMMAND [args]
```

Through the CronTab:

```
Min Hour Month Date Day "DMC COMMAND [args]"
```

It is possible to talk to DLGMail via ARExx, in an ARExx script, if you desire to do so. The port name is (case sensitive) DLGM_ARExx.

Commands Added to *.Startup Scripts

Please refer to the DLGMail command summary for exact syntax and use.

You will want to add one or two DLGMail commands to most of your DLG *.startup scripts to enhance the operation of your bulletin board system:

DELAYCALL Add this to the TR?.startup script which has TrapDoor

USEREXPORT Add this to ALL *.startup scripts, including the script for your local port. A typical *.startup file, from TPT, looks similar to this (toward the end)

```
DLG:Hangup
DLG:CronEvent >NIL: d "> NIL: DLG:REMOVEUSER TRx"
DLG:FreePort >NIL: -p TRx -k "BBS"
```

Between the CronEvent and FreePort lines, insert:

```
FIDO:DMC >NIL: USEREXPORT
```

In ONLY the startup file for the port with TrapDoor, follow the new line above with

```
FIDO:DMC >NIL: DELAYCALL SECONDS 25
```

Thus, the end of the modified *.startup scripts will look similar to this:

```
DLG:Hangup
DLG:CronEvent >NIL: d "> NIL: DLG:REMOVEUSER TRx"
FIDO:DMC >NIL: USEREXPORT
FIDO:DMC >NIL: DELAYCALL SECONDS 25
DLG:FreePort >NIL: -p TRx -k "BBS"
```

Causing DLGMail To Make Calls

Assuming you've configured AUTOCALL in your .CFG file to some number other than 0, and assuming that there is mail to be delivered, DLGMail will attempt to dial out every several seconds all by itself, while observing the CALLTYPES flavor mask.

The DLGMail commands CALL and DELAYCALL operate similarly, however are specifically programmed by you or executed from a script file to cause calls to be placed.

For more specialized needs, the following dialing commands also exist:

NORMAL This causes nodes with mail waiting as "normal" (.FLO and .OUT) to be dialed. Additionally, nodes with crashmail and file requests will also be called.

DIRECT This causes nodes with direct mail to be called. Nodes flavored crash and normal will also be called as well.

DIRECTONLY This causes nodes with direct mail to be called exclusively.

CRASH This causes nodes with crash (continuous) mail to be called.

REQUEST Causes nodes with file requests to be dialed.

Mail Processing Commands

The following commands are used to cause various mail processing activities to be started:

PROCESS Perform DLGImp, if necessary run DLGExp, DLGNet and DLGBundle.

EXPORT Perform DLGExp and DLGNet scans only, and export every area on the system. Process normally only exports the areas which were just imported, so it is important to use EXPORT at least once a day to gather up mail in any areas which didn't receive new mail on a given day. DLGBundle is also run if any packets are created which require bundling.

There are some options you apply as well:

EXPORT tagname tagname tagname

Causes only the list of echos, specified by tagname, to be exported. Netmail will also be scanned.

EXPORT area# area# area#

Causes only the list of echos, specified by DLG area number, to be exported. Netmail will also be scanned.

EXPORT 0

Causes the passthrough message areas to be scanned. Normally a useless endeavor, however if you have passthrough areas set up and your machine were to crash during the export, this would allow you to go back and finish exporting in those areas.

NETSCAN Causes DLGNet to scan for netmail, file attaches and file requests. DLGBundle will be run if any packets are created which require bundling.

CHANGEFLOW ARGS. This causes DLGMail to change the flavoring of flow files on the system, similar to

CHANGEPKT ARGS. which reflavors packets only.

TrapDoor Specific Commands (handled within DMC immediately)

TRAPDOOR OFF Disables TrapDoor from loading. Will cause TrapDoor to exit if TrapDoor is already running. Please note that this command utilizes TPT's GetPort. You may also utilize this command to unload TrapDoor so you can use a terminal program, provided you don't attempt to have your own GetPort try to lock TRx.

TRAPDUMP OFF Same as above, though this command will pend waiting for TrapDoor to become free so it can unload. This will tie up DLGMail's ARexx port, causing other commands to "back up" - this is not recommended.

TRAPDOOR ON Enables TrapDoor to load.

TRAPDUMP ON Same as above.

ZMH ON Reconfigures TrapDoor for Zone Mail Hour, preventing it from dropping to the BBS and preventing your system from honoring File Requests (FREQu).s).

ZMH OFF Reconfigures TrapDoor for normal BBS operation and FREQu.s).

TRAPDOOR NOANSWER Tells TrapDoor to answer on the 5000th ring, effectively disabling TrapDoor from answering the phone.

TRAPDOOR ANSWER Tells TrapDoor to answer on the 1st ring, effectively reenabling TrapDoor to answer the phone.

Nodelist Processing Commands

NODELIST DIFF Executes "FIDO:Batch/NLDiff.DMB," and this should be executed once a day to check for and automatically process the NODEDIFF file.

NODELIST NEW Executes "FIDO:Batch/NLNew.DMB." This should only be run to process a brand-new full nodelist (for instance, if you've missed one or more diff files).

NODELIST RECOMPILE Executes "FIDO:Batch/NLRecompile.DMB." Use this to recompile the nodelist after making any changes to TrapList.CFG.

Miscellaneous

RECONFIG Partially reconfigures DLGMail, rereading the DLGMail.CFG file. Most settings are altered, but it may be necessary to kill DLGMail and reinvite it under some circumstances. Please note that unless you kill then restart DLGMail, you'll need to RECONFIG each time you make a change in the config file!

REPORT Generates a LOGS:DLGMailConfig.RPT file, allowing you to view all the configurable (and a few non-configurable) settings of your DLGMail system.

QUICKATTACH Premise: You desire to attach files to one or several nodes and want to avoid the trouble of leaving a netmail message, the possible errors that can be created from this, etc.

For each file you wish to attach, you can send it to one or several other FidoNet nodes in one or several different ways.

QuickAttach accepts several arguments (up to 22 or a maximum CLI command line length of 256, whichever is smaller), and the format is:

DMC QUICKATTACH <path:filename> <mode> <dest1> [[dest or mode] [dest or mode] etc.]

Mode can be one of the following: CRASH, NORMAL, DIRECT, HOLD or EXISTING, and a selected mode will stay in effect until the next occurrence of a mode parameter. When EXISTING is used, QuickAttach searches for the first available flow file for the destination node and appends your new file attach to that. If no existing flow file is located, QuickAttach defaults to creating a NORMAL (.flo) file attach.

For instance, let's say you'd like to attach the file ODD:MYFILE to nodes 1/1, 2/2 and 3/3. You'd like to have it sent when rates are cheap, so you'd want to send it NORMAL. Your command line might look like this:

```
1> DMC QUICKATTACH odd:myfile NORMAL 1/1 2/2 3/3
```

Let's say that 3/3 will only accept the file during zone mail hour (which, if you haven't caught on yet, is when DIRECT mail is sent):

```
1> DMC QUICKATTACH odd:myfile NORMAL 1/1 2/2 DIRECT 3/3
```

Let's assume that node 1/1 is local and can accept continuous mail:

```
1> DMC QUICKATTACH odd:myfile CRASH 1/1 NORMAL 2/2 DIRECT 3/3
```

Let's assume that you normally poll 4/4 for mail and want to send that system a file, but don't want to call it unnecessarily. You'd want to use EXISTING, which would attach the file to the first flow file for that node it found, be it crash, normal, direct or hold. If no flow file exists, a NORMAL flow file will be created.

```
1> DMC QUICKATTACH odd:myfile EXISTING 4/4
```

Please note that QuickAttach will accept zone and point numbers in the destination spec, but zones are ignored at this time.

QUICKREQUEST ARGS. Allows a quick and easy way of requesting files other nodes. Usage is:

DMC QuickRequest <address> <file> [[file].]

A file request will be built for you.

REPORT Generates a LOGS:DLGMailConfig.RPT file, which will show you the various settings being used by DLGMail.

QUIT Unloads DLGMail. If you should use this, please remember that DMC will continue to queue events so if you restart DLGMail later, all queued events will be performed.

There may be other miscellaneous DLGMail commands which have not been fully explained in this section. Please refer to the overview elsewhere for these commands and their syntax.

DLGMail.MAC Macro File

It may be convenient to utilize the DLGMail.MAC macro file to hold all references to DMC commands which require extensions. For instance, let's say you "change flow and poll" a node once a night for mail, but sometimes you like to do it manually in the afternoon or at some other odd time. Instead of having a complete entry in the CronTab and also having to type the whole thing in at the CLI at odd times, you could create an entry in the macro file instead. As an example, let's say you need to do the following:

```
CHANGEFLOW 1:202/503.0 ? TO N POLL
```

You'd normally need to place this into your CronTab like so:

```
30 11 * * * FIDO:DMC CHANGEFLOW 1:202/503.0 ? TO N POLL
```

And if you wished to poll for mail at some odd hour, you'd normally have to enter this at the CLI:

```
1> DMC CHANGEFLOW 1:202/503.0 ? TO N POLL
```

Utilizing the macro capability, you'd instead create an entry in the DLGMail.MAC file something like this:

```
POLLMIKE CHANGEFLOW 1:202/503.0 ? TO N POLL
```

And in your CronTab you'd use:

```
30 11 * * * FIDO:DMC POLLMIKE
```

And when you wanted to poll Mike's system for mail at an odd time, from the CLI you'd simply:

```
1> DMC POLLMIKE
```

The .MAC file is arranged very simply. Each line contains a macro name, followed by whitespace, followed by the usual command/argument line as you would have used elsewhere. Obviously, the macro name may not contain any spaces. There is no limit to the number of macros you may define, and the names you use are up to you so long as no spaces are imbedded in the name itself. Case isn't important, therefore when creating names, observe that "MyName" and "MYNAME" and "myname" are all equivalent and the first occurrence of any of these will be substituted and used as the actual DMC command.

DLGMail Scripts

Anyone who previously had to fight with a gillion script files to use Fido with DLG (or any other BBS, for that matter) will be pleasantly surprised when using DLGMail - the only mandatory scripts are those used to initially start DLGMail and to process the nodelist! Some modifications to existing DLG scripts are required.

For those with special needs, "script hooks" have been added to DLGMail to allow the execution of batch files at certain times during DLGMail's run. Note that these are OPTIONAL and certainly are not required, but facilitate the use of additional Fido processing software, or to handle special (and oddball) needs.

DLGMail's operation is broken up into "modules" and within most of these modules you'll find the script hooks. All scripts are expected to be located in FIDO:Batch/. While DLGMail comes with a suggested layout including the FIDO: assignment, this is the only place where the FIDO: assignment is directly required.

"DLGMail_Startup.DMB"

This script is executed early in DLGMail's startup. It should be possible to place all DLGMail-related assignments into this script, and thus avoid the FidoStart.Batch file entirely (though please note: you won't be able to add any paths for CLI's using this method).

"DLGMail_Shutdown.DMB"

This script is executed when DLGMail closes down (either normally or from detection of an error) and is the very last thing done before DLGMail exits.

"ZMH_ON.DMB" and "ZMH_OFF.DMB"

These scripts are called after DLGMail reconfigures TrapDoor for normal/Zone Mail Hour operation. Please note that it is NOT generally necessary to call either of these, as TrapDoor is reconfigured adequately for most people (FREQu are forbidden during zone mail hour, etc., and reallowed during other times).

"PRE_Nodelist.DMB" and "POST_Nodelist.DMB"

These scripts are executed before nodelist processing begins (any method) and after nodelist processing is completed (any method). These are intended for people who need to enable/disable other utilities during nodelist processing.

"TrapDoor_OFF.DMB" and "TrapDoor_ON.DMB"

These are executed after TrapDoor is disabled/enabled, respectively.

"PRE_Process.DMB" and "POST_Process.DMB"

These are executed before and after the "Process Cycle" loop.

"PRE_Import.DMB" and "POST_Process.DMB"

These are executed before and after DLGImp is executed, within the "Process Cycle" loop.

"GenericTick.DMB"

This is executed IN PLACE OF PDQTick, and only if "FIDO:PDQTick" does not exist.

"PRE_Export.DMB" and "POST_Export.DMB"

These are executed before and after DLGExp, wherever DLGExp may be called (PROCESS, USEREXPORT, EXPORT commands may all call DLGExp).

"PRE_Netscan.DMB" and "POST_Netscan.DMB"

These are executed before and after DLGNet, wherever DLGNet may be called (PROCESS, USEREXPORT, EXPORT, NETSCAN commands may all call DLGNet).

"PRE_Bundle.DMB" and POST_Bundle.DMB"

These are executed before and after DLGBundle, wherever DLGBundle may be called (PROCESS, USEREXPORT, EXPORT, NETSCAN commands may all call DLGBundle).

"PRE_Accounting.DMB"

This is executed prior to clearing TrapList's call accounting variables. It is intended for using a script to log accounting variables, or with the use of other utilities or some very creative scripting, could change the flavor of packets or flowfiles to prevent your system from making calls to "dead" nodes.

"GenericTick.DMB"

If you have chosen the generic Tick option in your .CFG file, the script GenericTick.DMB will be executed during the process cycle whenever a .TIC file is detected in INBOUND.

Additional Support Executables

You absolutely must have ARC, PKXArc, ZOO, Zip, UnZIP, UnARJ and LHA in your command search path in order to use the default archiving and extraction strings built into DLGMail.

WARNING: It has been found that PKXARC (and it's brother, PKAX, we assume) will fail under certain circumstances on the Amiga 3000. For this reason, DLGMail no longer will use these alternatives to ARC in order to decompress bundles of mail. PKXArc is still used in processing nodelists and nodediff files.

WARNING: Though there appears to be no problem in using LZ to bundle LHA mail to other Amiga systems which utilize LZ, some MS-DOS systems have difficulty determining that LHA uncompression is to be used on mail bundles of that type and as such, it would be better to limit LHA mail to other Amiga systems using LZ at this time.

As you may have already guessed, DLGImp understands six archive types for incoming mail, ARC, ZOO, ZIP, LZH, LHA and ARJ. DLGBundle will allow you to build any of these bundles for outgoing mail (Contingent on your having a CLI-based ARJ in your path).

Observe that you can configure your system to utilize different archivers by configuring them in the DLGMail.CFG file.

DLGImp By Itself

Invoking DLGImp

DLGImp is the DLGMail Fido message importing program, intended to be called into action by the Fido manager, DLGMail. It CANNOT be invoked without DLGMail.

To cause DLGImp to import mail, use DMC like so:

```
1> DMC PROCESS
```

DLGImp Quick Overview

DLGImp is invoked only when DLGMail is commanded to PROCESS mail. DLGImp scans the INBOUND: mail directory looking for

Packets
Bundles

Packets will be processed first. Following that, bundles are uncompressed resulting in more packets to be processed. This "loop" is repeated until no packets or bundles exist.

If a routed packet is found, the configuration setting PKTRROUTE is checked. If set to 0, the routed bundle is deleted. If set to 1, the packet is physically moved to the OUTBOUND: directory. If set to 2, the packet is processed as if it were addressed to your system.

Linking (if the configuration setting LINKREPLIES is set to 1) is then done, where DLGImp will attempt to chain together all the new messages it just imported via the subject line in each message header. The linking process is rather low-tech in nature and is far from perfect, but most users and operators find the results adequate for their needs.

DLGImp then creates a file called DLGMail.EXP which lists all the areas (by tagname) it just imported.

DLGImp exits, informing DLGMail what types of mail were imported which identifies what additional processing is required (if any).

If netmail addressed to Areafix or ZAreafix is detected, a special list of these messages is built so that PDQAreafix can later be called to act on them.

How DLGImp Works

ARCmail which is being unarched is renamed to TMP_BUNDLE.XXX - it's examined to determine the type of archive, then renamed TMP_BUNDLE.xxx (where .xxx is the typical archiver extension). These files are easy to spot if you take a dir of your inbound:. As soon as the unarchiver is completed, the TMP_BUNDLE.xxx file is deleted, leaving only the *.PKT contents to be dealt with. This prevents a mail bundle from being tossed over and over and over should a problem arise.

A mail bundle, once unarchived, contains one or more xxxxxxxx.PKT files. The xxxxxxxx is a pseudo-random number (depending on what program named it), with the .PKT extension designating the file as a packet of mail (both net and echomail can be present).

Each *.PKT file contains a header with information regarding who it was intended for. Should, for some strange reason (or on purpose, of course), a system sends you a packet not destined for your system, it will be processed, moved or deleted as you have designated in your configuration settings.

DLGImp will also allow routed netmail (coming in to your system as individual messages instead of packets) to pass through your system if you have DLGNet set up to allow it. The CRASH bit on any routed netmail is stripped to prevent your system from incurring long distance phone bills to send other people's mail.

When tossing a message, DLGImp uses the MSG:XX/pointers.msg file to decide what to number the message. Thus, if the pointer file has 86 for a high message, the message about to be tossed will be 87. The pointers.msg file is then updated.

As DLGImp tosses messages, it also looks to see how many messages belong in an area. If, for instance, area 100 should have 50 messages in it and the pointers (before a toss) are (low) 50 and (high) 100, 51 messages exist at that moment. DLGImp knows that there should be only 50, so it deletes messages 50 and 51 (Message 50 to bring the area in-line with what should be there, message 51 to make room for the message it is about to toss). Then it will toss message 101 and store the new pointers, 52 and 101.

You must have previously set up DLGMail.CFG and DLGMail.ARE appropriately or DLGImp will fail miserably.

Dupe checking and cursory examination of messages for origin and seen-by lines is done during importing. If an echomail message fails any of these tests, or the echo area isn't configured on your system, the message will be tossed to the DLG area you've designated for BAD messages.

Two activity logs are maintained by DLGImp within each area. The first, "Import.DAT," is written to at the end of each import. It is a human readable ASCII text file which shows the date of creation, the date mail was last imported to the area, and the number of messages imported between those two dates.

The second, "Activity.DAT," is configurable and may be turned off to save space. The Activity.DAT file contains extensive information about the area, including the number of messages imported, exported, number of systems exported to and total messages exported on a day-to-day basis. The structure of this file will be made public, and utilities designed to analyze this file are being written and will be available from Intuitive Software.

DLGExp By Itself

Invoking DLGExp

Like DLGImp, DLGExp is intended to be launched by DLGMail. It will not run when invoked directly.

To cause DLGExp to export echomail, use DMC like so:

```
1> DMC EXPORT
```

How DLGExp works

A number of messages are deposited into an echomail area, either by a local user on your system entering it or by importing (using DLGImp or other processor) from some distant system far away.

When a message is entered on your system locally, obviously the origin of the message is known. Until DLGExp processes it, no other systems have either seen it or passed it through. Because the message was local, the originating node's address information is known.

When a message is imported, your feed for the message has his address stored in the header of the message, along with addressee, addressor, subject, date, etc. Next comes the message text, then the very last text line of the message (known as the Origin line) contains the originating node's address, plus perhaps it's name and may be some information about the sender's BBS. Depending on how you have your DLG message base configured, the SEEN-BY information is usually hidden

from view but can be displayed if desired and follows the origin line. PATH information follows the SEEN-BY list and is always hidden from general viewing. To review this information, one must use a text editor or file zapper.

For each echomail message area, a pointer called the Hi Water Mark is used by DLGExp and maintained by DLG itself (when the message base is renumbered) to constantly point at the highest message previously exported by DLGExp. The common method of storing the Hi Water Mark is to embed the pointer into the header of a standard message (MSG:<areanumber>/1.msg). DLGMail provides for this (for compatibility with other Fido processing utilities).

The file which maintains distribution information about all the echomail areas on your system is called DLGMail.ARE. See documentation about it elsewhere in the DLGMail documentation.

Normally, when DLGExp begins to export, it searches every echomail area in the DLGMail.ARE file (this operation can be altered with the DLGMail.EXP file but for purposes of our discussion now, will be ignored).

For each area, the following occurs:

First, it obtains DLG's low and high message pointers as well as the hi water pointer. It compares the high message pointer to the hi water mark and if the hi water pointer equals the high message pointer, the area is skipped.

If the hi water pointer is missing, DLGExp assumes that it should process all messages in the area. No real harm is done here, though the results can be unnerving for nodes who have recently connected to your system and get a batch of old messages.

Once a range of messages to process has been established (hi water +1 to the high message in the area), DLGExp begins to move through each message, examining the messages to determine:

- 1) Who has previously been sent a copy of the message (the SEEN-BY list)
- 2) Comparing it to the distribution fields in the areas.bbs file.

From this operation, DLGExp determines who to send the message on to. If a local message is being processed, then nobody in the distribution fields will have seen the message and the message will need to be processed for all nodes in the distribution list. If the message was imported and if there is more than one node in the distribution list, the message will undoubtedly need to be sent on to all but one. If there is only one node in the distribution list and the message was imported from that node, then the message can be skipped.

If the message is to be sent on to other nodes, the message is rewritten to disk with the additional nodes appearing in the SEEN-BY area. Your own address is added to the PATH information.

At this point, the message is added to separate destination outbound packets of messages in your OUTBOUND: directory. The format of these packed messages is slightly different from those stored on disk, and DLGExp takes care of this conversion. After the message is added to each destination packet, DLGExp moves on to the next message.

The distribution SEEN-BY and PATH information is normally processed into existing information when messages are added to packets, however if the destination node is in a different zone or the STRIP modifier is present in that node's entry in that area's entry in the DLGMail.ARE file, only your system's addressing information will be transferred with the message text.

After all the known messages in an area have been processed, the hi water marker is rewritten to reflect this, but the area is once again checked to see that users haven't entered any additional messages in the area while processing was taking place (ah! the magic of multitasking!). If not, DLGExp moves on to the next area.

For areas configured as passthrough, all messages are deleted once the area has been exported.

Once all areas have been processed, the OUTBOUND: directory will contain "raw" message packets which must be "closed" for processing later by a separate program (DLGBundle). To close these packets, now named in the format of zone.net.node.point.RAWPKT, the file is renamed to the format of xxxxxxxx.PKT (xxxxxxx is an eight digit pseudo-random number), and if the rename is successful then terminating zero bytes are appended to the file (notating to mail importers on other systems that the end of the packet has been reached). Since the destination is nondiscernable from a .PKT name, the destination information is retained in the packet's filenote.

At this point, DLGExp has done its deed and is through.

DLGNet By Itself

Invoking DLGNet

You likely know the drill: You can't run DLGNet by itself. To have DLGNet scan and process netmail, use DMC like so:

```
1> DMC NETSCAN
```

How DLGNet Operates

DLGNet retrieves the netmail area pointers and compares the high pointer to it's netmail high water marker. If new messages exist, DLGNet will then begin to process them. If no new messages exist, DLGNet exits.

If the high water marker file disappears, DLGNet will process the entire netmail area - this will usually result in nothing being packetted up except legitimately unprocessed messages, but if something has gone astray, there may be additional activity.

Each message for processing is examined to determine several key attributes:

The address the message is to

The address the message is from

Whether or not the SENT bit has been set

If the message has the CRASH bit set

If the message has the FILEATTACHED bit set

If the message has the FILEREQUEST bit set

If the message has the KILLSSENT bit set

If the message has INTL or TOPT kludges

NOTE: DLGMail assumes it will always be operated in a node environment. If your DLG/DLGMail system is a point off another node, results may be unpredictable (at best).

If the SENT bit is set, processing of that message ceases and the high water pointer is updated.

Next, the destination address is determined and checked against your addresses (including all AKA's). If the message is at its destination, processing of it concludes and the high water pointer is updated.

If the FILEREQUEST bit is set, a special dispatch to DLGMail's QUICKREQUEST routine is made, the high water pointer is updated, and the message itself is deleted to prevent a rescan from reprocessing the request again.

If the FILEATTACHED bit is set, a special dispatch to DLGMail's QUICKATTACH routine is made and processing of that message continues as normal.

At this point, the message needs to be processed..

e, all routing is skipped.

If file DLGMail.BUN is examined to see if the destination address
should take place.

If, the netmail-specific routing file DLGMail.MRT is examined.

will have been determined and a packet will then be opened. In
it will be of the form zone.net.node.point.RAWPKT. For
a packet called zone.net.node.point.CUT will be opened.

Added to the packet, the packet closed, the high water pointer
for netmail will take place.

Found for your system, if you have points enabled a file called
the message has been addressed to. If the name is found,
its point address.

More than one network where you have more than one address,
it netmail to a sister system which is not your primary address
on that particular message.

Added, the OUTBOUND: directory will be examined looking for the
id, the file will be renamed to a pseudo-random 8 digit
id with a pair of null bytes. The .CUT packets previously written
id and require no further attention.

to DLGMail on the status of packets created which will allow
necessary.

;

It by DLGMail, and DLGMail will only invoke DLGBundle when
if *.PKT files to be processed further.

s and to either:

create a new bundle and move them to it or append them to an

ed for *.PKT files. When one is found, the destination informa-
tion the DLGMail.BUN routing file is examined.

mation is sought (but is not necessary). After the "route" has
checked for (if no entry, ARC is assumed) then a transmission
DIRECT is assumed).

n.

n the *.pkt will be renamed or appended to a packet with the

t a bit more complicated.

e presence of any bundle name to the destination. If found, the
file will be created with an appropriate day/counter name and
. If the original bundle was not zero bytes in length, the packet

In the case of bundles, a flow file is created or appended to with the name of the bundle just processed. If the flow file already existed, the file is always checked to be sure that the bundle name is present.

Processing continues until no *.PKT files exist in OUTBOUND:.

Miscellaneous

Interrupting DLGMail

No break handling has been installed, therefore interrupting any of the DLGMail executables while it is running will leave memory allocated, some files open, etc. In a pinch, it's OK to break one (using the CLI 'break' command) but plan to reboot shortly after.

DLGMail Extras

A variety of "extra" programs will be supplied to you with DLGMail. Below you'll find descriptions and usage information in order to use them to their fullest extent.

KillTD overview

See the DLG documentation on port locking and getting to fully understand the use of this executable.

KillTD requires no mandatory arguments and simply sends an "Abort F" message to the default TrapDoor ARexx port. If you include an argument, KillTD will send the Abort F to an ARexx port with the name of the argument.

A replacement TD.Batch script has been supplied, but for documentation purposes you would replace the -b argument in TD.Batch with -b "FIDO:KillTD"

For systems with multiple TrapDoor front-ends, where each TrapDoor must have a different ARexx port name, you may utilize KillTD to kill each of the different ports.

KillTD defaults to using the port name:

"TrapDoor" if DLGMail isn't running, or

The DLGMail-configured TrapDoor port name if DLGMail is running

To specify a different port name other than default, follow KillTD with the name of the TrapDoor ARexx port:

```
1> KillTd port2
```

In this example, TrapDoor "port2" would be send an "abort f" and that TrapDoor would close down and unload.

Servicing Points

Points are "sub-nodes" off of your address. If your address is 1:114/52.0, your points would be addressed as 1:114/52.1 through 1:114/52.999 (any point number between 1 and 999 is acceptable to DLGMail). Please keep in mind that you are responsible to the network(s) for points just as you would be responsible for your users.

Configuration

You must configure DLGMail.CFG with:

```
POINTENABLE 1
```


and, if you must service points limited to 2D pointnet kludging, include the following AKA as your first AKA, plus enable the pointnet stripping:

```
POINTNETAKA 1:<pointnet>/0.0
```

To actually select a "pointnet" number isn't difficult. Pick an unusual looking network number and search the nodelist file looking for "host,<pointnet>" or, optionally, use the Trap* utility GetNode and search for 1:<pointnet>/0, 2:<pointnet>/0, 3:<pointnet>/0, etc.

If your search turns up no matches, you are free to use it, but keep in mind that while the odds are small, an actual net number could someday appear matching your pointnet number, so keep your pointnet somewhat unique.

You are limited to pointnet numbers greater than 0 and less than 32767, by the way.

Don't forget to put your AKA into TrapDoor's configuration file, though it doesn't need to appear as the first AKA listed.

In the 4D addressing scheme, the point's number is obvious. In the 2D pointnet scheme, the point's number is placed in the node number field, thus:

1:114/52.5 is equivalent in point use to 1:5252/5.0

In both of the above cases, the point would be number 5.

Feeding Points Netmail

Netmail fully addressed to points will automatically be forwarded to them. Configure the DLGMail.PNT file if you wish to allow Netmail to be forwarded to the point when previously it hadn't been addressed to the point properly.

Feeding Points Echomail

Simply add them to the distribution list in your DLGMail.ARE file, either as their 4D or 2D address. Observe the necessary requirements in the PDQMail.AFX file for 2D points.

Bundler Specifications

DLGMail.BUN file entries for your points are, at first, somewhat difficult to explain. Take for fact the following:

One "global" entry should be all that is required for 2D points, similar to this:

```
1:5252/*.*.0 ARCHIVER LZH TYPE HOLD
```

Several entries for your 4D points will be required, however:

```
1:114/52.1 ARCHIVER LZH TYPE HOLD
1:114/52.3 ARCHIVER LZH TYPE HOLD
1:114/52.6 ARCHIVER LZH TYPE HOLD
1:114/52.7 ARCHIVER LZH TYPE HOLD
```

From the illustration above, it should be clear that it is OK to use wildcards for 2D points in your bundler file but not for 4D points. Why? Simple, really.

When DLGNet processes netmail for your points, it will be processing 4D point addresses - when one of your 2D points receives netmail, DLGNet will search the .BUN file first looking for the 4D address - a wildcarded address such as 1:114/52.* will result in a match, and the netmail would be built into a packet and bundled for the 4D address, not the kludge address that it actually needs.

Instead, what must be done is this: For each point which can accept 4D addressing, create a separate entry as above in the DLGMail.BUN file. For each point which uses the 2D kludge, create entries in the DLGMail.MRT file like so:

```
1:114/52.2 THRU 1:5252/2.0
1:114/52.4 THRU 1:5252/4.0
1:114/52.5 THRU 1:5252/5.0
```

Later, when the bundler finds packets addressed to the 2D kludged points, the wildcarded 1:5252/*.0 entry in the .BUN file will be found and used to properly bundle and flow the mail.

Though not absolutely necessary, the examples above will place all your point mail on hold (points usually call you to pick their mail up) and bundle all packets to them using the efficient LHArc algorithm. If you need to bundle certain points differently, you can opt for adding specific entries for them earlier in the DLGMail.BUN file and still include the more global method above for ease.

Routing To/From Points

Mail and file attaches you send will be routed to your points normally by following the steps above and configuring as shown. In addition, inbcond file attaches MEANT FOR ANY OF YOUR POINTS which are accompanied by a conventional netmail message with the FILEATTACHED bit set and proper filename entries in the subject line will also be routed to your points. This is the single exception to the convention of not routing file attaches.

File attaches from your points to other nodes or points not on your system cannot be routed since it is never really known whether or not such a file attach could result in a telephone cost to you.

Configuring The Point

There are many considerations beyond the scope of this document to consider when having a point configure his software. Every point software package, no matter what platform, has its unique abilities and quirks that must be worked around.

The point should build origin lines with the full point address in proper form (1:114/52.5 is required, NOT 1:5252/5.0!).

The point should not generate the FMPT kludge for echomail.

The point need not strip its pointnet kludge (if used) from the seen- by or path lines, though these addresses MUST be 2D compatible.

For netmail purposes, the point software should be capable of creating the required kludge lines so that recipients can identify its origin.

As point use matures and additional DLGMail users provide feedback on the system's shortcomings, the entire system will be modified to handle points in a better fashion. Until everything is foolproof, monitor the seen-by and path lines in echo areas fed to points to insure that your pointnet addressing doesn't end up "out in the network" and that nothing else particularly unusual occurs.

Of particular interest to points: You cannot "rescan" an echo area for your points as this could result in reexporting messages to all points you feed. It is still acceptable to rescan echo areas, just remember that points will not ever receive any rescanned messages.

Configuration

The best for last, you might say.

Editing Scripts and Configs

There is quite a bit to edit, but most of the more tedious edits need be made only once to get the system up and functional.

Editing S:Dialog.Start

Begin by editing your S:DLG.Start script file, the idea here being to execute S:FidoStart.Batch in approximately the same place as in the _Dialog.Start example file. If you are just now adding TrapDoor, you'll need to edit the ActivatePort entries to bring TrapDoor up.

Editing S:FidoStart.Batch

Here is where the various assignments needed for DLGMail are made, and where paths to those directories are added. Create your own FidoStart.Batch file from the example (leave the example intact!) and edit to suit your needs (and the directories you just created).

It is very important to edit ONLY those items in this file which you are directed to. Try to bring the rest of your system in-line with this file for optimum layout and performance (and fewer other things to have to edit later hidden in a jungle of configuration files).

Editing DLGBatch:TRx.Startup

Use the supplied example to add the "FIDO:DMC DELAYCALL SECONDS 25" entry to the startup file on the port with TrapDoor (usually TR0). Add the "FIDO:DMC USEREXPORT" line to ALL your TRx.startup and TLx.startup script files."

Editing the DLGBatch:CronTab

It is best for now if you skip this step and come back to it later, when you have a feel for how the system works together with DLG. Modify your existing CronTab file using the supplied example.

Editing FIDO:Batch/NL *.DMB Files

No editing should be necessary at all. Look them over to see what they do, however. NOTE AT THIS TIME THAT THE ARCHIVER CALLED PKXARC IS USED TO PROCESS THE NODELIST. Without modifications to the scripts, it's impossible to use ARC or PKAX. PKXARC was supplied to you, however, so this shouldn't be a problem.

Editing MAIL:Trap* Files

Dig out the TrapDoor and TrapList documentation, use supplied examples as a starting point, and begin modifying.

Editing FIDO:DLGMail.CFG

Of the five configuration files left, all DLGMail, this one should be edited first. With this manual in hand, examine each entry in the example and create a similar entry in your own configuration file.

There is no replacement for the time-honored tradition of studying the manual and getting your configuration right the first time. A little extra time spent now will save you much grief later.

IMPORTANT: If you are setting up for Fido for the first time and do not yet have a node number, you'll need to configure EVERYTHING to a temporary fictitious number in order to send the Net Coordinator in your area an "I'm here and would like a node number please" message as required in FidoNet "Policy 4" (the network's operational policy bible). Pick a fictitious number that resembles a real number for your area. Zone should be appropriate for the area of the world you are in (Zone 1 is North America, for instance). Determine from another operator in your immediate area the proper net number (net 114 is Phoenix, for example). In a pseudo-random fashion, choose a temporary node number to use - 9999 is usually safe, and your point should be 0 - therefore, for this example, your temporary address would be 1:114/9999.0.

It is an excellent idea to set PKTRoute to a value of 2 while awaiting notification of your new node number.

Editing FIDO:DLGMail.ARE

Refer to this manual to create one of your own, very simple in nature, to get you started. As you enter FidoNet and begin to pick up echomail, you'll need to edit this file on a continuing basis, so refer to the example as needed.

Editing FIDO:DLGMail.MRT

Create an empty file for the time being. Later, you'll add to this as necessary, but you need to get your feet wet first.

Editing FIDO:DLGMail.BUN

Indeed, all you need here is an entry for your net coordinator that looks something like this (be sure to get HIS address right - The most important number is the NET number (in the example, 114). Node 0 is always an alias to his real address, so 0 here is safe.):

```
1:114/0.0 ARCHIVER ARC TYPE CRASH
```

Refer to this manual for details on this file.

Configuring DLG and TrapDoor

Please refer to the DLG instructions for adding FidoNet configuration. If setting up for Fido for the first time, tell DLG you are the same temporary node you told TrapDoor and DLGMail.

The underscored Trap*.CFG files should help you out quite a bit in setting your Trap* executables. Refer to the documentation often, and don't forget to have copied the "traplist.library" file to your libs: directory or results won't be as you expect.

Refer to THIS chapter on incidentals that you'll need to get started.

Compiling the Nodelist

The first thing you need to do is to obtain a file called "NODELIST.ARC" - you may need to use a terminal program to download it beforehand. Copy this file into INBOUND:, and from the CLI type:

```
1> DMC NODELIST NEW
```

"DLGMail will execute the FIDO:Batch/NLNew.DMB script to copy the archived nodelist file into your NODELIST: directory, unarchive it, delete the archive, then call TrapList to process it. If all is successful, TrapList will leave several files in NODELIST:, notably:"

NODELIST.xxx The actual nodelist, over a megabyte in size
FidoNet.index An index into the nodelist
FidoNet.extra Additional special information that TrapDoor uses

If you have previously unarced the nodelist archive, you can copy the unarchived nodelist into NODELIST: and use a different command to begin processing:

```
1> DMC NODELIST Recompile
```

The difference here is that the steps involved in copying and unarchiving are removed.

Trying It Out

Since all the temporary configuring should have been done at this point, you are in a position to test the system out. Begin by logging onto your BBS and entering your netmail area.

Enter a message to the DLGMail support system. E to enter a message, answer Yes for private, address the message to Steve Lewis, subject doesn't really matter, and the following for FidoNet address information:

```
Zone: 1
```

Net: 114
Node.Point: 52

The system should, at this point, ask you, at this point, ask you if you wish to make this a file request, a file attach, or crash message. If it does not and immediately throws you into the editor, escape the editor, don't save the message, and enter the SysOp user editor and edit your own account to give yourself SysOp netmail privileges.

Try entering the message again, this time after entering the address information you should get the three questions pertaining to file attaches, requests and crashmail. Answer NO to file request, NO to file attach, but YES for crashmail.

Drop a few lines into the message, and save it as usual.

As soon as the message is saved, pop to your Workbench screen and watch - DLGMail's window will pop open, and DLGNet will process the message you just entered. The window will pop closed.

Congratulations! You've just created your first outgoing packet of messages (in this case, only one message). Take a directory of your OUTBOUND: directory and you'll find a lonely packet name: 1.114.52.0.CUT.

Almost immediately after you entered this message, DLGMail will have processed it and will attempt to have TrapDoor call and send this message to the DLGMail support board. You will observe some activity in your DLGMail status window, and if enabled, your DLGMail processing window. A few seconds later, DLGMail will tell TrapDoor to call 1:114/52.0.

Observe in the TrapDoor windows the results of this.

If the connection is successfully completed, your DLGMail will send the message; if unsuccessful, DLGMail will continue to attempt to call and deliver the message as many times as required so long as 1:114/52.0 is busy or doesn't answer. In the case that the systems do connect but transmission can't be completed for some reason, delivery will only be attempted approximately 4 times before TrapDoor will refuse to call again. See the TrapDoor documentation on call accounting for details on this.

Requesting a Permanent Fido Address

Assuming this test message made it OK, you may wish to try other test messages to other nodes or go directly to sending the NC (Net Coordinator) a request for a node number. You should have already downloaded FidoNet Policy 4 and understand what you need to tell the NC in your message to make you eligible for receiving a node number. Follow the policy to the letter, and include all requested information in your message.

Once his reply is received and you've read the message, he'll either have assigned you a node number or given you a reason why not, with instructions on what to do next.

If he has assigned you a permanent node number, re-edit every configuration on your system to reflect the permanent number - at minimum, this will require editing the DLG FidoNet configuration, TrapDoor.CFG, DLGMail.CFG, and possibly other DLGMail configuration files.

Depending on when you've requested and he's granted your node number, it will take from a few days to two or three weeks for your node to appear in the nodelist. At that time, anyone in the world will be able to netmail you. In the meantime, you'll still be able to netmail them.

Requesting an Echomail Hub

After receiving your node number, you'll want to netmail the Net Echomail Coordinator (NEC) and explain that you are new, just received your node number, and are requesting an echomail hub from which you can request echomail. Be sure and let the NEC know the capabilities of your modem as this may determine who he has feed you. The NEC should reply with a hub address and name and

any pertinent net policies regarding distribution. Get in contact with your new hub, preferably by voice if you manage to get a voice number, and acquaint him to your setup and capabilities to ease his mind - most hub operators will appreciate this extra step.

Receiving non-backbone (privately distributed) echos is similar - once you have a node number that hits the nodelist, almost anyone who carries the echos you desire will be willing to feed you, though you'll learn by trial and error who makes a good feed and who does not. In the case of the privately distributed echos, and unlike the distribution of the backbone echos, you'll likely have to make your own long distance calls, so plan accordingly. The DLGMail support board carries, in our humble opinion, the better private echos and all DLG-related support echos and will be happy to provide you with a very reliable feed. In addition, DLG and DLGMail upgrades and bug fix files are available there to automatically be file attached to you whenever new ones become available, so you'll want to explore receiving a feed from there.

Refer to previous sections in this manual for setting up new echo areas in the DLGMail.ARE file, and adding any new feeds to the DLGMail.BUN file.

Finding Your Comfort Zone

As with anything new, there's a lot to learn when adding Fido to your bulletin board system. Take it slow and easy, but don't be afraid. You'll find that most (not all, however) fellow FidoNet operators are helpful and courteous as they all went through the learning curve you have only just begun to climb onto.

Your fellow DLG/DLGMail operators will be the most helpful with any questions that arise on the operation and configuration of DLGMail, of course, and you'll find them anxious to help you.

Support from TPT and Intuitive Software is, of course, available.

Copyright

DLGMail is Copyright 1990, 1991, 1992 Intuitive Software (a collaborative effort between Steve Lewis and Ross Martin) and is under license to TelePro Technologies.

Nodelist Compiling

Introduction

This document is a brief description of TrapList, TrapDoor's fast Nodelist Processor, explaining how to set up the software in order to compile FidoNet compatible Nodestats and apply Nodediffs to them.

Please read the entire manual before using TrapList.

TrapList reads FidoNet compatible Nodelist and Nodediff files and produces an index and an extra file as outputs. These files are later used by TrapDoor and DLG to access the actual Nodelist in a fast way and to obtain important data from there, such as telephone numbers and session passwords.

Basics

TrapList is a CLI based application, it can't be run from WorkBench. It accepts a wide range of configuration statements that allow an easy and accurate configuration. It should be run using the 'DMC NODLIST' commands.

On invocation through a standard commandline interface or shell, TrapList will parse its configuration file, look for nodelist and nodediff files, apply matching nodediffs, parse the final nodelists, sort the nodes, delete duplicate entries and finally, write out an index as well as an extra file, both of them to the directory specified in the NODLISTPATH configuration statement.

The Nodelist

The Nodelist is a file that contains important data about all the nodes in FidoNet. It is often called "the glue which holds the network together". It is FidoNet's "phone book" and it defines the top-level network structure.

The Nodelist is published as an ASCII text file named NODLIST.nnn each Friday, where nnn is the day-of-year of the Friday publication date. This file is packed into an standard archive file (using System Enhancement Associates' ARC file format) named NODLIST.Ann, where nn are the last two digits of day-of-year.

You can skip the next chapter if you want. It is just here to supply you with additional background information about the format of the nodelist. You may want to come back to it when you need to build your own private nodelists, such as for a private point network.

The Nodediff

With more than nine thousand nodes as of this date, the nodelist, even in archive form, is a substantial document (or file). Since distribution is via electronic file transfer, this file is NOT routinely distributed. Instead, when a new nodelist is prepared, it is compared with the previous week's nodelist, and a file containing only the differences is created and distributed.

The distribution file, called NODDIFF.nnn, where nnn is the day-of-year of publication, is actually an editing script which will transform the previous week's nodelist into the current nodelist.

For actual distribution, NODDIFF.nnn is packed into an archive file named NODDIFF.Ann, where nn are the last two digits of day-of-year.

Configuration

Configuration statements are given to TrapList by means of a configuration file. This file can either be called "TrapList.cfg", or, more generally, "fido.cfg". The configuration file will be first searched for in the current directory, then in "MAIL:". You can override this search sequence and specify your own alternate configuration file using the CONFIG commandline keyword, such as in

```
TrapList CONFIG DEVS:traplist-configuration
```

For detailed information on the global configuration file "fido.cfg", please consult additional documentation about the general structure of this file, how several applications can share configuration keywords, etc.

Please note that the configuration statements are neither case sensitive nor do they have to appear in any specific order. Only one statement should appear on a single line.

Configuration Statements

BUFFERSIZE buffersize

TrapList will buffer all file i/o to speed up the nodelist processing time. This statement allows you to specify the amount of memory TrapList will use for each of its buffers. This value should be given in bytes. TrapList will always use at least buffers of 4096 bytes, even if a lower value is specified with BUFFERSIZE. The default is 32768 bytes.

```
Example: BufferSize 65536
```

(NO)CHECKCRC

When this configuration statement is used, TrapList will check the crc of the freshly generated nodelist when applying a diff. If the crc is not what it is supposed to be, TrapList will issue a warning, delete the new nodelist and continue to use the old one instead. That way, you can always be sure to have a valid nodelist.

```
Example: CheckCRC
```

COMMENT comment-identifier

When the nodelist is parsed, and TrapList encounters a comment line, it will compare the comment-identifier ("S" for Sysop comments, "U" for User comments, a.s.o.) with the identifiers you specified here. If it finds the identifier listed, and the REPORT keyword is in effect, the comment line will be written to the reportfile (see REPORT).

```
Examples: Comment SU      ; list "Sysop" and "User" comments
Comment A      ; list "All" comments
Comment SUFA    ; list all comments
```

COST costs dial-prefix [dial-prefix...]

Specify some phone costs per minute for a group of telephone numbers. When TrapDoor reads a phonenumber from the nodelist, and you have defined a certain cost, TrapDoor will be able to show you the total cost of your call. If you specify minus one ("-1") as the costs for a number, TrapDoor will never call there. You can use this to prevent TrapDoor from calling "-Unpublished-" numbers. See the example configuration file for more details. You can specify multiple dial prefixes for a single cost entry.

```
Examples: Cost 67 "43-1-" ; 0.67 ATS/min for local calls
Cost 1333 "30-" "44-" ; Greece and UK cost the same
Cost -1 "-" ; -Unpublished- = undialable
```

Cost 2800 "" ; 28 ATS/min for all others

(NO)DELOLDDIFFS

This tells TrapList to delete old nodediffs when they have been applied successfully.

Example: DelOldDiffs

(NO)DELOLDLISTS

This tells TrapList to delete old nodelists when they have been "diff'd" into a new one.

Example: DelOldLists

DIAL original-string replacement-string

Specify a dial translation for phone numbers. When a phonenumber is read in from the nodelist by TrapDoor, these dial translations will be applied. If the beginning of the telephone number matches the original-string, it will be replaced by the replacement-string. Dial translations will be searched from first to last, and only the first match will be applied. After the first match, no further translation will be done.

Examples: Dial "43-1-" "" ; strip country and exchange code
; for local calls
Dial "43-" "0" ; replace country code with "0"
; for the rest of Austria
Dial "-" "-" ; do not translate "-Unpublished-"
Dial "" "00" ; add "00" to other numbers

(NO)GENNEWSTYLE

Generate the "new style" index files, consisting of a fidonet.index and a fidonet.extra file. This type of index is used by TrapDoor 1.60 and above. A shared library called "traplist.library" is used to access these "new style" files.

Example: GenNewStyle

NODELIST nodelistname [DIFF diffname]

This specifies the name of a nodelist file to look for. TrapList will append day numbers to the filename and search for the nodelist with the highest number. If the DIFF keyword was specified also, TrapList will try to locate a matching nodediff file and apply it to the nodelist.

If a node is listed in more than one nodelist (for example, in the main FidoNet nodelist and in a private pointnet), the data from the first nodelist will be taken (in the order that the nodelists are listed in your configuration file). This allows you to override data for a node in the nodelist; just list the system in your private nodelist and specify the name of the private list before the main nodelist in the configuration file for TrapList.

Note: Nodelists will be scanned in reverse order than specified. This is not a fault.

Examples: Nodelist "NODELIST" DIFF "NODEDIFF" Nodelist "POINTNET"

NODELISTPATH path

This (required) configuration statement tells TrapList where to find Nodelist and Nodediff files and where to put its index files. It is not necessary to terminate the path specification with a slash or colon.

Example: NodelistPath "Nodelist:"

PASSWORD password node [node...]

Associate a password with a certain nodenumber. TrapList will put the password into the extra file and TrapDoor will then automatically use that password when it establishes communication with that node. Also, on incoming calls, TrapDoor will check the password for the calling node and hang up, if incorrect.

If you set up a password with a node that is also listed as a Zone Coordinator, Regional Coordinator, Net Coordinator or Hub Coordinator, make sure you specify all of the node's addresses, or you might experience strange problems when calling one of the Coordinator's addresses. Note that you may specify multiple node addresses for a single password, especially for that purpose.

There also is a special password for nodes you want to disable from connecting to your system, "—NONE—". You may want to use this keyword for your own addresses, thus forbidding any caller from (fraudulently) using your address. TrapDoor will immediately disconnect if anyone calls you with the given address.

If you generate "old-style" index files, please note that the password should not be longer than 6 characters. If you happen to use longer passwords, TrapList will display a warning and truncate the password after the sixth character. Actually, passwords could be up to eight characters, but for compatibility reasons (CList) ...

```
Examples: Password "secret" 2:310/6 2:3160/0
Password "geheim" 2:310/3
```

REPORT reportfile

Write some statistics about the processed nodelists into the given reportfile in addition to displaying them on stdout (the shell window). In conjunction with other utilities, such as Electric Herald, this can be used to automatically send the sysop information about freshly compiled nodelists and similar things. If you use the COMMENT keyword, selected comment lines from the nodelist will also be sent to the reportfile.

```
Example: Report "Mail:Report.txt"
```

ZONE zone

This is the default zone that TrapList should use for nodelists that do not contain a Zone keyword, i.e. Pointnets and similar private lists.

```
Example: Zone 2
```

Example Configuration File

```
; *****
; *** Sample TrapList configuration file ***
; *****
GENNEWSTYLE
;change this to your zone number;
Zone 1
BufferSize 65536 ; use 64k buffer for file i/o
CHECKCRC
DELOLDLISTS
DELOLDDIFFS
REPORT LOGS:TrapList.LOG
NODELISTPATH "NodeList:" ; where to look for nodelist files
;NodeList "pointnet" ; my private pointnet
NodeList "NODELIST" Diff "NODEDIFF" ; the weekly St.Louis nodelist
; this is an example password entry
Password SECRET 1:114/52
```



```

;these are example dial string conversions
Dial "34-" "01132-" ; Spain
Dial "43-" "01143-" ; Austria
Dial "44-" "01144-" ; UK
Dial "39-" "01139-" ; ITALY
Dial "49-" "01149-" ; Germany
Dial "46-" "01146-" ; Sweden
Dial "61-" "01161-" ; Australia

;these are Saskatoon local exchanges. You must replace these with
;the local exchanges in your area. A list of these exchanges
;is normally available from your local net coordinator.

Dial "1-306-221-" "221-" ; Sask Tel Cellular in Saskatoon
Dial "1-306-239-" "239-" ; Osler|
Dial "1-306-241-" "241-" ; Cantel cellular in Saskatoon
Dial "1-306-" "1-"

; this entry is needed to catch anything that did not match above ;
Dial "1-" "1-"

; Cost "43-222-" 67 ; local calls
; Cost "43-" 100 ; would be nice

; These are example cost entries. They are for the Saskatoon area
; Once again, cost tables for your area are normally available
; from your local net coordinator.

Cost 0 "1-306-221" ; Sask Tel cellular in Saskatoon
Cost 0 "1-306-239" ; Osler
Cost 0 "1-306-241" ; Cantel cellular in Saskatoon

```

Processing The Lists

To process the latest nodelist and nodediff, simply call TrapList with "TrapList" or by using the DMC NODELIST commands. Using the DMC NODELIST commands will not give you any output but if you invoke Traplist directly from the CLI you should get an output quite like this:

```

TrapList 1.32 N The TrapDoor Nodelist Processor
' Copyright 1990, 1991 by Martin J. Laubach
and Maximilian Hantsch
All rights reserved

Applying NODEDIFF.054 to NODELIST.047

Parsing NODELIST.054

Zone 2 Europe
Zone 1 North_America
Zone 3 Oceania
Zone 4 America_Latina
Zone 5 AFRICA
Sorting
Processing
Writing
Total Nodes Processed      = 7360
Bad Nodes                  = 0
Unique Nodes               = 7360
Nodes Deleted (Down)      = 119
Points                     = 0
NNNNNNNNNNNNNNNNNNNNNN

```

```

Zone Coordinators Listed   =    5
Regional Coordinators Listed =   38
Network Coordinators Listed =  300
Hub Coordinators Listed   =  405
=====
Diff'ing                   =  0:27
Parsing                   =  0:19
Sorting                   =  0:06
Processing                 =  0:02
Writing                   =  0:05
NNNNNNNNNNNNNNNNNNNNNN
Total time needed         =  0:59

```

Return Values

If everything went all right, and the index and extra files were both correctly produced and renamed, TrapList will return zero (0).

If index and extra files could be correctly generated, but not renamed to their final names (or the old ones not deleted), TrapList will return 10. In that case, terminate all programs that still have locks open on the nodelist index/extra files, then execute:

```
cd nodelist: ; or wherever your nodelist files are
```

The following lines are for the new format. ; the "fidonet.extra" file is automatically updated ; via the "traplist.library". delete fidonet.index.t rename fidonet.index.t fidonet.index

On any other error, TrapList will return 20.

Note that in case of an error, the old index and extra files will be left unchanged, so TrapDoor will still be able to access a valid nodelist, even though it will not be the latest one in such a case.

The following names are either trademarks or the efforts of the person and/or company listed:

- Amiga and AmigaDOS are trademarks of Commodore-Amiga, Inc.
- ARC by System Enhancement Associates.
- Chameleon Editor and CList by Jürgen Hermann.
- Fido and FidoNet are trademarks of Tom Jennings, Fido Software.
- License Agreement inspired by Jack Radigan and the GNU General Public License.
- Many Thanks to our Alpha and Beta testing crew, including, but not limited to: Arnout Grootveld, Carl-Christian Kanne, Heiko Richter, Johannes Mistelbauer, Manfred Schödler, Peter Wlcek, Renè Hexel, Telepro Technologies, Tony Miller ...

Nodelist and Nodediff Information from FTS-0005.

Special Thanks to all our Registered TrapDoor Users.

TrapDoor by Maximilian Hantsch and Martin Laubach.

TrapList by Martin Laubach and Maximilian Hantsch.

TrapDoor

Introduction

TrapDoor is DLG's FidoNet-compatible front end mailer. It transfers mail from/to other FidoNet-compatible systems. When FidoNet is being run on your DLG system, TrapDoor will take the place of the DLG's SETUP module and answer all incoming calls passing control over to the BBS when necessary. Because of this, some of the modem settings in the port configuration module will be rendered inactive and will be replaced by modem settings in your "TrapDoor.cfg" file.

Installation

Upon invocation, TrapDoor will try to find a file "TrapDoor.cfg" in the current directory or in "MAIL:". This file should contain your default configuration data for TrapDoor, like your system's name, your nodenumber etc.

Basics

This section is intended to give you a quick overview about the operation of TrapDoor and certain selected topics. If you wish to look up a specific command keyword, please see the reference section "Configuration Commands".

FidoNet

FidoNet is a world-wide network of many "FidoNet-compatible" bulletin boards, which communicate with each other using "mailers". TrapDoor is one of these mailers, it allows you to send and receive FidoNet mail and files.

If you are not familiar with FidoNet, or certain terms used in this manual, please see "TrapDoor's FidoNet Manual" (FidoNet.Man located on Disk3) and the glossary there.

Mailer Operation

Trapdoor transfers mail bundles and/or files to/from another FidoNet system. DLGMail prepares the outgoing mail and files in the "Outbound: directory" and will then tell TrapDoor to call other FidoNet systems as required. TrapDoor will use the modem to place a telephone call there, then proceed to transfer mail/files with built-in protocols. TrapDoor will also receive mail and files that are waiting at the other system for you. These incoming data will be stored in the "inbound: directory", from where you (or the mail tossing software) can further process the mail and files.

Nodelist

Nodelist: is a directory listing all (or parts of) the nodes in FidoNet. It stores the names of systems and sysops, the node numbers, and the telephone number for each node. TrapDoor can use the nodelist to look-up telephone numbers before placing an outgoing call.

A separate program called "TrapList" is used to process the nodelist before TrapDoor can reference it. TrapList builds index files for the nodelist, which TrapDoor and DLG later use. For more information on TrapList and how to compile/index nodelists, please see the chapter on TrapList.

Compatibility

A sensitive point in every FidoNet mailer is "compatibility". FidoNet uses at least three different handshake methods and more than three different transfer protocols. Not every mailer implements all of them, and so it is very important that mailers automatically detect the capabilities of the other system and switch to the correct handshake and protocol. This detection phase is also one of the sources of most errors and failures to establish a proper connection between two FidoNet systems.

TrapDoor has been extensively tested with other mailers, including various versions of BinkleyTerm, D'Bridge, FrontDoor, Paragon and TrapDoor itself. It should work fine with most systems.

There are two additional notes about compatibility, however. The EMSI handshake method sometimes causes problems with some BBSs with an integrated mailer, such as Opus or Paragon. Should you repeatedly experience strange hangups with such a system, you should disable EMSI using the NOEMSI keyword and try again. Especially older version of Paragon are known to not work correctly with EMSI. If you regularly call such a system, you might want to set up a "custom configuration" for that system (see the next section).

Second, the FTS-1 "Lotek" protocol is very often poorly implemented in other mailers. TrapDoor uses a very strict version of this protocol, and behaves exactly according to the specifications. Other mailers with errors in their FTS-1 protocol code fail with TrapDoor. Two known cases are Tabby, a Macintosh mailer, which fails completely, and TIMS, an MS-DOS mailer, which fail to transfer the names of inbound files. TrapDoor will name the files "an unnamed file 001", "an unnamed file 002" etc.

Custom Configuration Entries

Sometimes it is desirable to use a special custom configuration when calling a certain node. For example, if you regularly connect to a system which cannot handle EMSI, you will want to disable EMSI when calling that node.

TrapDoor allows you to set up such custom configuration entries. There is a utility called "setconfig" to store a configuration string for a certain node number. For example, to disable EMSI for 2:314/471, enter

```
setconfig 2:314/471 "NOEMSI"
```

Accounting

TrapDoor can keep a count of all outgoing calls and incoming calls, on a per node basis, and the cost for those calls. When you have enabled accounting (see the (NO)ACCOUNTING keyword), TrapDoor will maintain a database of

- the number of outgoing calls to a node
- the number of incoming calls from a node
- the number of successful sessions with a node
- the number of failed sessions with a node
- the number of BUSY signals when calling a node
- the number of NO CARRIER signals when calling a node
- the number of VOICE signals when calling a node
- the total cost of all outgoing calls to a node

The "listacct" tool will display all the accounting information, and "clearacct" will reset it.

By using the ACCTMAX keyword, you can specify a maximum for all of the above accounting items. You can set limits, and TrapDoor will refuse to call a node when at least one of the limits is exceeded. For example, you could set the maximum for the number of failed sessions to 2, and TrapDoor will refuse to call a node again after two failed connects (where a connection was made, but the carrier was lost before the successful end of the session).

Security

It is often important to make sure that a particular system is really who it claims to be. For that reason, TrapDoor can use passwords to protect mail sessions. You can set up a password for each node you regularly exchange mail with, and TrapDoor will check the password every time this node calls you. If the password which the other end sends and the password that is stored in the local database do not match, TrapDoor will hang up, or, if the password for an AKA address is wrong, it will simply "forget" that AKA and will continue the session without sending mail or files for the AKA address.

The "setpasswd" tool allows you to set up or remove passwords. For example, to setup a password of "vienna" for 2:313/28, enter

```
setpasswd 2:313/28 "vienna"
```

TrapDoor will always compare passwords using a case-insignificant match. It will, however, send out passwords exactly as you typed them, in case the other end does not use a case-insensitive compare.

To protect you from losing your password database in case of a system crash, you can also enter "Password" statements in your TrapList configuration file and have TrapList set up all your passwords. This is described in more detail in the chapter on TrapList.

If you are running a private system, and you do not want to receive calls from any other nodes except the ones with which you have established passwords, you can set up a secret password using the PASSWORD keyword. Other mailers will only be able to call your system if either (a) they know the secret password or (b) you have set up another password for them (using setpasswd) and they send the latter one.

If you are running a public FidoNet node, do not use the PASSWORD keyword.

File Tagging

Another security feature of TrapDoor is "file tagging". Each received file will be tagged with a "Secure" field in the filenote, if it was received either from a node listed in the nodelist or from a node with which a password was set up. The "Secure"-Tag will contain either "NL" or "PW" or both "NL,PW" depending on the particular security measures under which the file was received. This can be used by a mail tosser to only automatically toss mail from password-protected sessions, for example.

Also, every file received will be tagged with the node number of the system from which it was received, in a "From" field.

Every FidoNet session will be assigned a unique number. This number will be recorded in the log file, and all files received in this session will be tagged with a "Trx" field listing this unique "transaction id".

If files were renamed during the receive operation (for example, because the file already existed), the original filename will be stored in another field in the filenote, tagged as "FileName".

Refusing and Pickup Preferences

The Emsi handshake allows for "Refusing" and "Pickup Preferences". These features work only when an Emsi handshake was chosen at the beginning of the session.

Refusing means that a system may refuse to receive certain files or certain mail packets at specific times. A system may refuse

- file requests
- compressed mail
- file attaches
- all of the above

TrapDoor will respect these wishes and not send the appropriate items, unless NOALLOWREFUSING is in effect.

Pickup Preferences means that another system may choose what mail or files to pick up from TrapDoor. A system may want to pick up mail and files

- for all presented addresses
- for the primary address only (no AKAs)
- no pickup at all

TrapDoor will respect these preferences and only send the requested items. (NO)ALLOWREFUSING does not affect this behaviour.

At the moment, there is no way to set the Refusing/Pickup mode in the EMSI packet that TrapDoor sends to the remote system. TrapDoor will always pick up mail for all presented addresses, and will never refuse to receive anything. This may be improved in a future version.

The Keyboard

TrapDoor features sophisticated keyboard handling, including the ability to assign arbitrary configuration commands to function keys.

One of the most important keys is probably the ESC (Escape) key. Pressing this key during a session will abort it. TrapDoor will hangup as soon as possible and either return to answer mode, or, if the call was not initiated from answer mode, exit. When TrapDoor is idle and you press the ESC key, it will reset the modem.

Next, TrapDoor makes use of the Alt (Alternate) key for system functions. When TrapDoor is idle and waiting in answer mode, you can activate a number of things via Alt-key sequences. The Alt key works like a shift key, you have to hold it down while pressing another key. These are the Alt key sequences that TrapDoor understands:

Alt-A ... immediately Answer the phone.

Alt-C ... reread the Config file "MAIL:TrapDoor.cfg". This is useful if you have changed the config file and want to reset TrapDoor to the new settings.

Alt-Q ... Quit, same as Alt-X.

Alt-R ... Reset modem.

Alt-S ... toggle Showrexx mode (see SHOWREXX keyword).

Alt-X ... eXit, same as Alt-Q.

If you get stuck and cannot remember a certain key, just press HELP:

HELP pop up the Help display.

Please note that TrapDoor will wait for the Help window to close before it exits.

From the main help screen, you can select "Settings" or "Keys". "Settings" will give you a short summary of all the current settings of your TrapDoor, "Keys" will show you a summary of all standard key assignments of TrapDoor. Select "Done" when finished reading. From the "Keys" display, you can also choose "F-Keys" which will show you a summary of the current function key assignments.

Function key assignments can be put on any of the function keys (reasonable, eh?). These are the keys in the topmost row of your Amiga keyboard, beside the ESC key. The FKEY keyword is used to assign function keys. When you press a function key, and there is an assignment for it, TrapDoor will send the string assigned to that key to its own ARexx port. From there, the assignment will be processed. That way, you can put things like "Call 2:310/6" or "NOEMSI" on your function keys, or you can start external programs with the "Run" and "Spawn" keywords.

Always remember that hitting a function key will cause TrapDoor to send an ARexx-message to itself. Thus, it will only process that keystroke when it is idle in answer mode. (It will actually process the keypress immediately, but the ARexx-message will be waiting at the ARexx-port until TrapDoor reads it when it comes back to the "Waiting for call" state.)

Trapdoor Setup

The first step in setting up your TrapDoor configurations is to study the TrapList section and compile the FidoNet nodelist. (If you do not have a nodelist yet, you should use TrapDoor or a terminal program to obtain one from another FidoNet node.)

The next step is to configure all of the information in your MSG:_Trapdoor.cfg file using your favorite text editor and rename it to remove the _ at the beginning of the name. Details for configuring this file are included later on in this chapter.

The main idea in a node setup is to run TrapDoor in answer mode. It will accept incoming calls directing them to the BBS or exchanging mail. While it is waiting for a call it is also waits for ARexx commands.

The Modem

The Modem must be Hayes compatible. Other modem command standards are currently not supported.

A word about cabling: you will need a seven-wire RS232C cable, that is one that supports at least RxD, TxD, CTS, RTS, DTR, DSR, DCD and last but not least, Gnd. Anything less (or other connections) may not have the desired effect, although the strict requirements which TrapDoor poses on the DSR signal can be bypassed by use of the NODSR keyword. Also, seven wire handshake can be disabled with NO7WIRE.

Be sure to set up your modem so that dropping DTR causes a hangup, and a return to command state — even better a complete reset. If you don't do this, TrapDoor will be incapable of hanging up correctly! On almost all Hayes compatible modems this can be achieved with AT&D3, a few may require changes to their DIP switch settings or S-registers. Please consult the manual that comes with your modem if you feel unsure.

Also, take care that your modem should respond to successful connects with a "CONNECT XXXX" message (where XXXX is the baudrate), not just "CONNECT" (except at 300 baud). If your modem returns only "CONNECT", TrapDoor assumes that the connection takes place at 300 baud. On most Hayes compatible modems you will have to use ATX1 or higher. If your modem returns "CONNECT FAST" (Trailblazer modems do), TrapDoor will continue to operate at the baudrate specified in the BAUD statement.

When DTR is set high, DSR should follow. In case your modem is reacting too slow, try adjusting the SLOWMODEM parameter. If your modem cannot properly handle DSR at all, use the NODSR setting.

Some modems require a substantial delay between the "AT" prefix and the actual command string. If this is the case with your modem, put at least one tilde ("~") character in between the "AT" prefix and the command. This will cause TrapDoor to wait a short time before sending the rest of the string. Fine tune the time with SLOWMODEM. Some modems also require a substantial delay after a reset (caused by DTR drop or ATZ) before they respond to commands again — insert tilde characters where appropriate!

Example Modem Settings

US Robotics Courier HST Modems

These are the settings for a Dual Standard HST modem. If you have the HST only or V.32 only version of the Courier, just set the parameters that apply to your modem only.

USRobotics Courier 14400 HST Dual Standard NRAM Settings...

```
DIAL=PULSE B1 F1 M3 X7
BAUD=19200 PARITY=N WORDLEN=8

&A3 &B1 &G0 &H1 &I0 &J0 &K3 &L0
&M4 &N0 &P0 &R2 &S0 &X0 &Y1

S02=043 S03=013 S04=010 S05=008
S06=007 S07=060 S08=002 S09=006
S10=007 S11=070 S12=050 S13=001
S15=008 S19=002 S21=010 S22=017
S23=019 S24=150 S26=000 S27=000
S28=008 S38=000

STORED PHONE #0:
#1:
#2:
#3:
```

To setup your modem, enter a terminal program. Select the baudrate at which you want to "lock" your modem, usually 19200 baud. (Warning: The serial.device of the Amiga - up to AmigaOS 1.3.3 - can not keep up with 38400 baud. Unless you have a faster processor, like a 68020 or 68030, you will get lots of transmission errors if you choose a higher baudrate than 19200.) Then type

```
AT&FM3X7&A3&B1&H1&K3&R2S13=1S15=8&W
```

and press return. Your modem is now set up for use with TrapDoor.

In your TrapDoor.cfg file, use the lines

```
BAUD 19200 LOCK
7WIRE
SLOWMODEM 10
MODEMINIT ""~ATZ|"
MODEMHANGUP "|^"
MODEMDIALPRE ""~ATB1DP"
MODEMDIALPOST "|"
MODEMANSWER ""~ATBOS7=30A|"
```

Ordinary 2400 Baud Modems

Set your modem to:

```

E1 echo modem commands
Q0 display result codes
V1 verbose results
X1 or higher
&C1 DCD follows carrier detect
&D3 reset on dtr drop
&S0 DSR always on

```

To setup your modem, enter a terminal program and set the baudrate to 2400 baud. Then type

```
AT&FE1Q0V1X4&C1&D3&S0&W
```

and press return. Your modem is now set up for use with TrapDoor.

In your TrapDoor.cfg file, use the lines

```

BAUD 2400
SLOWMODEM 10
MODEMINIT ""AT~Z|~"
MODEMHANGUP "|^"
MODEMDIALPRE ""AT~DP"
MODEMDIALPOST "|~"
MODEMANSWER ""AT~A|~"

```

Internal Supra 2400zi Modems

Set your modem to:

```

E1 L2 M1 Q0 V1 X4 B0 Y0
&C1 &D3 &G0 &J0 &L0 &M0 &P0 &S0
S0=0 S2=43 S3=13 S4=10 S5=8 S6=2 S7=20 S8=2 S9=6
S10=14 S12=50 S25=5 S26=1

```

To setup your modem, enter a terminal program that can talk to "modem0.device" and set the baudrate to 2400 baud. Then type

```
AT&FE1Q0V1X4&C1&D3&S0&W
```

and press return. Your modem is now set up for use with TrapDoor.

In your TrapDoor.cfg file, use the lines

```

SERIALNAME "modem0.device"
BAUD 2400
SLOWMODEM 10
MODEMINIT ""AT~Z|~"
MODEMHANGUP "|^"
MODEMDIALPRE ""AT~DP"
MODEMDIALPOST "|~"
MODEMANSWER ""AT~A|~"

```

Configuration Commands

Modem Commands

All the modem commands (MODEMINIT, MODEMDIALPRE, MODEMDIALPOST, MODEMANSWER and MODEMHANGUP — all explained later on) accept a few special characters in the configuration string. These are:

```

~ ... short delay
^ ... drop DTR, wait a while, raise DTR again
| ... send a carriage return character

```

All other characters will be sent to the modem unchanged and without any further action.

Embedded Percent-Commands

The AFTERSESSION, BBSCOMMAND, DIALER and FREQUENT keywords allow you to specify a command-string that will be executed. In these command-strings, the following embedded %-commands are permitted. All of them are case-sensitive.

%b baudrate (between computer and modem)
%B baudrate (of the actual connection)
%s serial device name
%u serial device unit number
%f serial device flags
%r unique random number (timestamp)
%l name of logfile
%Z zone number of the other system
%N net number of the other system
%F fido node number of the other system
%P point number of the other system
%n complete fidonet address of the other system
%% %

For the FREQUENT keyword, the following sequences are allowed in addition to the above:

%i name of the .REQ file (input file)
%o name of the .RLO file (output file)
%S name of Sysop of other system

Please note that %b and %B are equivalent unless you use LOCK. In that case, %b will reflect the LOCKed baudrate, whereas %B will give you the baudrate of the actual connection (that was returned by the modem in a "CONNECT XXXX" message).

Keywords

(NO)7WIRE

Enable/disable 7-wire cabling. This will instruct the serial device to use (or not to use) the CTS and RTS signals. If you are using high speed modems with data compression, such as a US Robotics Courier HST, you must use 7WIRE handshake and you must set your modem to "hardware handshake" mode, otherwise various difficulties will arise.

Examples:

```
7Wire
No7Wire
```

ABORT signals

Abort is the ARexx equivalent of ^C, ^D, ^E and ^F. To "simulate" such a keypress, just send an "ABORT k" message, where k can be any of C, D, E or F. Multiple signals are OK, ie. "ABORT CDEF" works as intended. For example, to terminate a TrapDoor that runs in answer mode, use "ABORT F".

(+) ARexx only command

(@) asynchronous execution possible (@ABORT)

Example:

```
TrapTell "Abort F"
```


(NO)ACCOUNTING

Turn accounting on or off. When enabled, TrapDoor will keep track of the number of calls made to a node, the total cost of all outgoing calls to that node, the number of calls received from that node, the number of successful and failed sessions as well as the number of calls that failed because of BUSY, NO CONNECTION or VOICE modem result strings.

Example:

```
Accounting
```

ACCTMAX limits

When accounting is enabled, TrapDoor, when instructed to call a certain node, will check whether or not this node exceeds the accounting limits set by this keyword.

The limits parameter, which must be given as a single string, sets

- the maximum costs,
- the number of calls out,
- number of failed sessions (where a connection could be established, but the carrier was lost before the successful end of the session),
- number of "busy" results,
- number of "no connection/no carrier/no dialtone" results,
- number of "voice" results,

in that order. To disable a certain count, just set it to minus one.

Example:

```
AcctMax "100 50 10 -1 -1 0"
```

sets the limits for further outgoing calls to

- Total cost thus far ≤ 100 .
- Number of calls made ≤ 50 .
- Number of failed sessions ≤ 10 .
- Number of BUSY, NO CONNECTION doesn't matter.
- Number of VOICE ≤ 0 (ie. don't call when there was VOICE only once).

ADJUST factor

Unfortunately, the Amiga serial.device software has a small problem with baudrates: Not only will it calculate the value to stuff into the baudrate register of the serial hardware incorrectly and therefore use baudrates that are a bit offset from the correct value, but also this behaviour is different on NTSC and PAL machines, which makes it even worse.

Some modems will work fine with such slightly wrong baudrates, others will not tolerate this and give a lot of transmission errors. By the way, this - it seems - is the main reason why programs such as BinkleyTerm Amiga fail to work with high-speed modems.

TrapDoor offers you a cure for such problems: ADJUST allows you to specify how much TrapDoor will vary any given baudrate before it passes it on to the serial.device. This value should be given in thousands (1/1000). An example: at the default value of -11, a baudrate of 2400 will be adjusted to $2400 - 1.1\% = 2400 - 26.4 = 2373.6$ baud. This value will be rounded to an integer and passed to the serial.device, which will then miscalculate the values for the hardware registers and set the hardware to almost exactly 2400 baud.

Normally, you should leave this parameter at the default value. If you have a HST or similar buffering modem, you can try to set it to zero. If you happen to live in the US (or any other country using 60 Hz video systems), you will probably have to set it to zero, or maybe even something else. Experiment!

This strange misbehaviour of the serial.device may be corrected in the next version of the Amiga operating system, in which case you should use a value of zero for ADJUST.

Example:

```
Adjust -11
```

AFTERSESSION command-string

As soon as TrapDoor finishes a FidoNet session with another mailer, and hangs up the modem, it will call the command you specify here. There may be embedded %-commands, which will be substituted by the parameters of the session that just ended; a list of them can be found in the chapter "Embedded Percent-Commands". To turn off the AFTERSESSION command, use AFTERSESSION "" or omit the statement completely.

Examples:

```
AfterSession "FIDO:DMC Process"  
AfterSession ""
```

AKA akalist

During the EMSI handshake, not only your main address, but also a list of other "also-known-as" addresses will be sent to the other system. Using the AKA keyword, you can specify all your AKA addresses. <akalist> should be a single string, listing all your other addresses.

There is a limit of 20 AKAs for your system.

Example:

```
Aka "2:3160/0 2:31/0 27:47/11"
```

(NO)ALLOWREFUSING

Enables (disables) AllowRefusing mode. Default is ALLOWREFUSING.

When AllowRefusing is enabled, TrapDoor respects the other end's wishes in the EMSI handshake. When the other end says it doesn't want to receive any compressed mail, for example, TrapDoor won't send it. With NoAllowRefusing, TrapDoor will always send what has to be sent.

Examples:

```
AllowRefusing  
NoAllowRefusing
```

ANSWER

Forces TrapDoor to operate in "answer mode". TD will then wait for a call, answer the phone and try to start a session with the remote system. While in answer mode, TrapDoor will accept commands via its ARexx interface.

Example:

```
Answer
```

(NO)AUTOOVERSCAN

When using the SCREENMODE CUSTOM or SCREENMODE TRAPDOOR option, switching this on forces TrapDoor to open its screen not to the standard NTSC or PAL size, but to the maximum size of the WorkBench screen. If you use a program like "MoreRows" to expand your workbench screen into overscan regions, and you would like TrapDoor to do the same, use this parameter.

Examples:

AutoOverScan
NoAutoOverScan

(NO)BACKGROUND

When BACKGROUND is turned on, TrapDoor will open its screen behind all other screens. Also, when the log and status window are opened, they will not be activated, so your currently activated window will stay activated.

BANNER bannerline

This line will be sent to the other side when TrapDoor answers a call. This typically identifies your system, says "hi", or tells human callers to hang up.

Instead of a single line, TrapDoor can also send a text file. In that case, use "<filename" as your BANNER (where filename should indicate your banner file). Note that your banner file should not be too long. A few lines will suffice. TrapDoor will automatically convert LFs to CRLFs when sending the file.

Examples:

```
Banner "TrapDoor Development [HST], online Mon-Sun 00:00-06:00"  
Banner "<mail:banner.txt"
```

BAUD baudrate

This is the baudrate to initially talk to the modem — after power-on or a reset. This speed may change during a session, when you did not lock the baud rate (see LOCK) and a different speed is reported by the modem.

Example:

```
Baud 2400
```

BBSCHAR character

To allow for other "drop-to-the-BBS" keys, especially useful for users on machines that do not have an ESC key (some Macintosh models, C64). The ascii value you indicate here will be recognized and treated as if it were ESC.

There are three ways to specify the character:

- decimal ASCII code: just specify the decimal digits of the ASCII code
- hexadecimal ASCII code: use a dollar sign ("\$\$") followed by the hexadecimal ASCII code
- the ASCII character itself: either prepend it with a single quote ("'"), or use just the character if it does not conflict with the other options (such as the dollar sign).

Examples:

```
BBSChar 35  
BBSChar $21  
BBSChar !  
BBSChar '!'
```

BBSCOMMAND command-string

This is used to set the command line that TrapDoor will execute to start a BBS. There may be embedded %-commands, these are described in the chapter "Embedded Percent-Commands".

When a human caller presses ESC, the %-commands will be replaced with the corresponding value and then the resulting string will be executed.

BBSCOMMAND will only be used if the BBSMODE is set to SPAWN or EXIT.

Example:

```
BBSCOMMAND "immed TR0 %b %B -w"
```

BBSINOUT device

This can be used to set stdin and stdout for the BBSCOMMAND to something else than those of the initial CLI from which TrapDoor was started. Just set it to any valid AmigaDOS device, like AUX:

To switch off this feature, use BBSINOUT "".

Example:

```
BBSInOut AUX:
```

BBSMODE mode

There are four modes available for connecting a BBS to TrapDoor:

NONE ... There is no BBS, TrapDoor will display "Mail only system — please hang up" to human callers.

SPAWN .. TrapDoor will use the BBSCOMMAND setting to start the BBS when a human caller enters an ESC character. When the command returns, TrapDoor will reinitialize the modem and continue to wait for a call. For this mode you must include the -w option in your BBSCOMMAND line. There is a corresponding configuration setting in your DLGMail config.

EXIT ... TrapDoor will start the BBSCOMMAND just as with SPAWN, but as soon as the command returns, TrapDoor will exit. For this mode you must exclude the -w switch in your BBSCOMMAND line. There is a corresponding configuration setting in your DLGMail config.

ZMH There is a BBS, but at the moment, human access to the BBS is closed due to "Zone Mail Hour". TrapDoor will display "Mail only period — please call later" to human callers.

Examples:

```
BBSMode Spawn  
BBSMode None  
BBSMode ZMH
```

BOSS zone:net/node.point

For a NODE setup, this should be set to your own address.

Specifies the FidoNet address of your boss, as in "2:310/3". Please be careful as not to leave out the zone and point information when your boss node is capable of four dimensional addressing.

Note: The setting of BOSS also specifies what mail will be sent to the other system when you call out using a telephone number. So, if you are calling 2:310/3 with "TrapDoor call 0043-1-454330" to request a file, you have to set BOSS to 2:310/3 for that call, too.

Example:

```
BOSS 2:310/3
```

CALL number | fido-address

The CALL parameter is generally not used in a DLG setup but is described here anyway.

Tells TrapDoor to call out. The number dialed can either be set in the configuration file with another CALL statement, or given directly in the command line: when the string following the CALL keyword is not BOSS, it will be interpreted as the number to call. Otherwise, the number from the configuration file will be used.

If you have enabled nodelist support (see NODELIST), you can also specify a FidoNet node number instead of the telephone number. Please note that you always have to prepend a zone number to the node number, as TrapDoor will use the zone-separating colon (":") to distinguish node numbers from phone numbers. When TrapDoor detects that you have given it a node number, it will consult the nodelist to find out the telephone number, the password and the baudrate (unless LOCKED) for that node and use these settings.

Examples:

```
CALL 1:140/90
CALL 1-234-5678
```

COLORS palette-specification

When you use SCREENMODE CUSTOM or SCREENMODE TRAPDOOR (see SCREENMODE), you can change the colors with this option. The palette specifier looks rather like a window specification, starting with color 0 (the background color) and continuing to color 3. The value for each color is given in decimal, using the formula

$\text{color} = \text{red} * 256 + \text{green} * 16 + \text{blue}$

where red, green and blue specify the intensity of each color (0 is none, 15 is highest intensity). As an alternative, each value can also be given in hexadecimal notation, if prepended with a dollar sign ("\$").

Examples:

```
Colors 2730/0/2560/10
Colors $aaa/$000/$a00/$00a
```

CONFIG config-file

This is not normally needed in a DLG environment. It allows you to put yet another config file in your favorite config directory in addition to the standard "TrapDoor.cfg" that TrapDoor looks for. The format of such a file is just the same as that of the command line — only that linefeeds will be ignored. Also, you may have comments embedded in the config file. Just precede them with a semicolon (";"), and TrapDoor will ignore the rest of the line starting at the semicolon.

Please note that several config files chaining one to the other are quite possible — but you may have to increase your stack size when you have a try at this. Also note that a recursive config file (ie. calling itself) is a rather bad idea since you may not have set your stack size to plus infinity.

Example:

```
Config Mail:TrapDoor.cfg
```

CREDITS

Displays some "About" information. Read this first — it will tell you a bit about this program and its authors.

Example:

```
Credits
```


DIALER command-string

Use this if you have a custom dial-out program that will perform special PCP-dialing, for example. If a string other than "" has been specified and TrapDoor is about to call out, instead of sending a dial command to the modem, TrapDoor will call the external command. When the external program returns, TrapDoor will immediately start the session handshake, just as if the IMMEDIATE option had been used.

To turn off the special DIALER feature and use the built-in modem handling, use DIALER "" or omit the statement completely. There may be embedded %-commands, these are described in the chapter "Embedded Percent-Commands".

Examples:

```
Dialer "callpcp"  
Dialer ""
```

(NO)DIETIFNA

Enable (disable) DietIfna mode. Choose this, when you only transfer small files or when the line quality is rather poor since it might then be faster than ZModem.

Examples:

```
NoDietIfna  
DietIfna
```

(NO)DSR

After TrapDoor opens the serial.device, it will wait a short while (depending on SLOWMODEM) and then sample the DSR line. This line should be activated when a valid data set (i.e. a functioning modem) is connected to the serial line. If TrapDoor doesn't find DSR activated, it will report "! modem not ready" and abort.

There are a few modems that cannot properly handle DSR. For these modems, use the NODSR setting. Note that when using NODSR, TrapDoor can't tell whether the modem is powered-on, on-line and ready.

Also note that if you have not connected DSR to the modem (if you have a wrong/bad cable), RTS/CTS handshake might not work correctly. This is due to the way the Amiga serial.device handles things.

Examples:

```
NoDsr  
Dsr
```

(NO)EMSI

Turns the EMSI handshake on or off.

Note that the EMSI protocol is rather new. Although EMSI is designed to be backwards compatible to older mailers, some fail when presented an EMSI handshake packet. If you experience any session handshake failures with other mailers, try again with EMSI disabled (use the NOEMSI switch).

If you regularly call a node that cannot handle EMSI, you can set up a custom configuration string for that node. See the chapter "Custom Configuration Entries".

Examples:

```
NoEmsi  
Emsi
```

FKEY function-key-assignment

Assign a configuration command string to a function key. The keyword must be followed by a single string, starting with the number of the function key (F1=1, F2=2,...; Shift-F1=11, Shift-F2=12,...), followed by a colon (":"), followed by the assignment.

If you wish to include spaces in the assignment, the argument must be enclosed in double-quotes; to include quotes within the quoted string, use the backslash ("\") as an escape character.

Examples:

```
FKey "1:Emsi"  
FKey "11:NoEmsi"  
FKey "5:Call Boss"  
FKey "6:Run CE"  
FKey "7:Run \"Execute Scripts:Import\""  
FKey "8:Run \"TrapPoll +r\""
```

FLOATLOCK string

HST modems support a mode (&B2) in which they will drop the DTE baud rate when a connection is made without error-correction, but will keep the baud rate locked on a connection with error-correction. FLOATLOCK allows you to use this mode.

Whenever TrapDoor reads a CONNECT xxx string from the modem, it will try to find the Floatlock string in the CONNECT result. If it finds it, it will keep the baudrate locked, otherwise it will switch to the baudrate found in the CONNECT string. You have to have LOCK enabled when you wish to use FLOATLOCK.

To turn off FLOATLOCK mode, use FLOATLOCK "" or omit the statement completely.

NOTE: We do not recommend to use FLOATLOCK. Instead, set your HST to &B1 and S15=8. This will provide slightly higher throughput on non-ARQ connections than using &B2 and FLOATLOCK.

Examples:

```
FLOATLOCK "/ARQ"  
FLOATLOCK ""
```

FREQUEST command-string

Sets the command to be executed as a file-request server on received file requests from remote systems. There may be embedded %-commands, these are described in the chapter "Embedded Percent-Commands".

The called command should then read the remote's request from the %i file, perform any action that it wants to do, and write a list of files that it wishes to send to the other side (including directory path) to the %o file.

To turn off the file-request capability, use FREQUEST "".

Examples:

```
FRequest "dlg:TPTFREQ MAIL:Files.txt %i %o"  
FRequest ""
```

(NO)IMMEDIATE

If you specify IMMEDIATE, TrapDoor will not care about modem commands, dialing, resetting the modem etc, but will go directly and immediately to the session handshake (EMSI, YooHoo or TSynch). So, if you have two Amigas connected via a serial cable, you can use TrapDoor to transfer files between them.

On the first machine, start TrapDoor with

```
TrapDoor ANSWER IMMEDIATE NODSR
```

TrapDoor will start up, open the serial device, ignore the DSR line (which has not been set high at that time because the other computer has not opened its serial device yet) and wait for approximately 30 seconds for a session handshake.

Now, on the other computer, run

```
TrapDoor CALL BOSS IMMEDIATE
```

TrapDoor will open its windows, and within a few seconds you should see the banner line of the other system appear in the status window. TrapDoor will then go on as usual with the session handshake and the actual file transfers.

INBOUND inbound-mail-directory

This should point to the directory where incoming files will be put. Usually "Mail:Inbound". Make sure that the TrapDoor inbound directory is the same as the one for your mail tosser, or your attempts to import mail will fail miserably. Additionally, TrapDoor uses this directory for temporary files during receiving and for files that store information about aborted/interrupted file transfers, so that receiving these files can be resumed in the next ZedZap session. For more information, see chapter "Inbound Directory".

Examples:

```
Inbound "Inbound:"
```

(NO)INTERLACE

Specify whether TrapDoor should open its screen in the interlace mode or not.

Examples:

```
NoInterlace  
Interlace
```

(NO)LOCK

Lock (do not lock) the baudrate. If the baudrate is locked, and TrapDoor receives a "CONNECT XXXX" message from the modem, TrapDoor will not adjust the baud rate to XXXX, but continue to operate at the rate specified with the BAUD keyword. Use this for buffered modems that convert the baud rates internally (for example, HST). Be sure to also configure the modem for a locked baudrate in that case (i.e. on HSTs, use &B1).

Examples:

```
NoLock  
Lock
```

LOGFILE filename

Sets the name of the logfile.

Example:

```
Logfile "Mail:TrapDoor.Log"
```

LOGLEVEL group:level

Sets the amount of logging information to be put into the logfile. There are ten logging groups, each of which deals with a different part of messages (such as modem, outbound, session protection, file transfers, general messages, external messages etc.). Each group maintains its own loglevel, which can range from 0 to 7.

The groups:

- 0 Debugger ? debugging messages, internal errors
- 1 Link - line is too bad, baudrate too low
- 2 Transfer | receiving xyz.txt, cps rates
- 3 System ! deleting file xyz.txt, user break,
- 4 Modem ~ NO CARRIER, BUSY, RING, VOICE
- 5 Session = Begin of session, picking up mail, session aborted, giving mail to, session connect time, cost
- 6 Security * bad password, unlisted system, node is undialable
- 7 Outside x spawning dialer, executing aftersession, spawning bbs
- 8 Information : sysop, name, aka, place, flags, using, trxid
- 9 Scheduler + waiting for call, incoming call detected, calling node

The levels:

- 0 Silent minimum logging
- 1 Terse terse logging
- 2 Discreet normal logging
- 3 Verbose detailed logging
- 4 Talkative extensive logging
- 5 Excessive very much logging
- 6 Annoying even more logging
- 7 Monologue maximum logging

Two (Discreet) seems to be a rather nice logging level and is the default for all groups. You might want to turn on more detailed logging in the Security and Information groups.

Examples:

```
Loglevel 5:3
Loglevel 4:4
```

LOGWINDOW window-specification

Use this keyword to change the position and size of the log window that TrapDoor opens. The window specification looks rather similar to a normal AmigaDOS CON:, RAW: or NEWCON: specification, but omit the device name and the window name. The correct format is: LeftEdge/TopEdge/Width/Height i.e. something like 0/20/640/150

Example:

```
LogWindow 0/20/640/150
```

MINBAUD baudrate

Minimal baudrate to establish a connection. Connections at baud rates below this limit will not be allowed, no matter if incoming or outgoing. TrapDoor will hang up immediately if the baud rate is lower than the value specified.

Example:

```
MinBaud 1200
```

(NO)MAXBAUD

If turned on, the MINBAUD parameter will automatically be set to the baudrate of the called system (found in the nodelist) or the own baudrate (set with BAUD) on outgoing calls, whichever is lower.

What is this good for? Let us assume you have a 2400 baud modem, and you wish to call a node that is also capable of 2400 baud. His nodelist entry also says he can do 2400 baud. Now, if you call him, sometimes, when line noise appears just when the modems negotiate the connect speed, this may cause a connection only at 1200 baud, or even worse, at 300 baud. If you have MAXBAUD enable, TrapDoor will immediately hang up in such a case.

On the other hand, if you are using an HST modem, and you are calling a system with a Trailblazer PEP modem, you might have problems when you use MAXBAUD: If the nodelist entry for the PEP modem specifies 9600 baud, TrapDoor will hang up if the connect speed is lower than 9600 baud. But: Standard HST modems and PEP modems use a different high speed protocol and can only talk at 2400 baud to each other. So, TrapDoor will hang up every time you call such a system.

You have to decide whether the usage of MAXBAUD is appropriate for you or not. You can also set up custom configuration entries to turn off MAXBAUD for selected nodes. See the chapter "Custom Configuration Entries" for more details.

Examples:

```
NoMaxBaud
MaxBaud
```

MODEMANSWER modem-answer-string

Modem answer string like "AT~AI". For special characters like "~", "^" and "I" that are allowed in the string, see chapter "Modem Commands". The total length of the string may not exceed 40 characters.

This string is sent to the modem whenever TrapDoor detects a "RING" and wants to answer the phone.

Example:

```
ModemAnswer "ATA|"
```

MODEMDIALPRE modem-pre-dial-string

Modem dial string such as "AT~DP" or "AT~DT". For special characters like "~", "^" or "I" that are allowed in the string, see chapter "Modem Commands". The total length of the string is limited to 40 characters.

This string is sent to the modem whenever TrapDoor wants to dial a number. After sending this string, TrapDoor will send the telephone number to dial, followed by the MODEMDIALPOST string.

Example:

```
ModemDialPre "ATDP"
```

MODEMDIALPOST modem-post-dial-string

Modem dial string such as "I". For special characters like "~", "^" and "I" that are allowed in the string, see chapter "Modem Commands". The total length of the string is limited to 40 characters.

This string is sent to the modem after the telephone number, when TrapDoor wants to dial a number. Also see the description of MODEMDIALPRE.

Example:

```
ModemDialPost "|"
```


MODEMHANGUP modem-hangup-string

This string will be sent to the modem whenever TrapDoor wants to hang up the line. There are many methods to accomplish this, including the strange "~~~~~ATHI" method. We do not recommend this. If you have configured your modem according to the descriptions in the chapter "The Modem", it should hang up as soon as DTR is lowered. So, the recommended hangup string is "|^|".

For special characters like "~", "^" and "|" that are allowed in the string, see chapter "Modem Commands". The total length of the string is limited to 40 characters.

Example:

```
ModemHangup "|^|"
```

MODEMINIT modem-init-string

This string will be sent to the modem during the initialization phase of TrapDoor. Things like "AT-S7=20I" can be done here. For special characters like "~", "^" and "|" that are allowed in the string, see chapter "Modem Commands". The total length of the string is limited to 80 characters.

TrapDoor will also check if it gets any response from the modem after sending this init string. The modem should at least send something like "OK". In fact, any carriage-return terminated string from the modem will suffice.

If TrapDoor is unable to detect a response from the modem, it will report "Initializing modem failed" and exit.

Example:

```
ModemInit "ATZ|"
```

NAME board's name

Name of this system — to be sent to the other system during the beginning of a mail session. The length of the string is limited to 60 characters.

Example:

```
Name "The Mad House Ñ TrapDoor Development"
```

NODE zone:net/node.point

Sets your own FidoNet address.

For points:

If your boss uses a mailer that is not point smart, be sure to insert your private pointnet number here and not your full address. In this case, you should not use your complete four dimensional address (2:310/3.24), but rather the fake pointnet addressing method "2:3000/24". FrontDoor, D'Bridge and newer versions of BinkleyTerm on the other hand do already support the four dimensional addressing method. Use your full address then.

For nodes:

Just set this to your own address.

Examples:

```
Node 2:3000/14
Node 2:310/3.14
Node 2:310/6
```

NODELIST nodelist-directory

Set this to the directory where you keep your nodelist files. TrapDoor understands only nodelist files generated by its own nodelist processor, TrapList. TrapDoor now uses the "traplist.library", which should be in your LIBS: directory, to access the nodelist.

To disable the nodelist support in TrapDoor, use NODELIST "".

Examples:

```
Nodelist "Nodelist:"  
Nodelist ""
```

OUTBOUND outbound-mail-directory

This should point to the directory where outgoing files are kept. It should contain all the necessary #?.REQ, #?.FLO, #?.HLO and #?.CLO files and the associated mail bundles. TrapDoor will automatically maintain and delete these files as they get sent out. Usually set to "Mail:Outbound". For more information, see chapter "Outbound directory".

Example:

```
OutBound "Mail:Outbound"
```

PASSWORD password

Specifies the password to be used for mail sessions.

On an outgoing call, if your password does not match the password that the other system has set up for you, you will be disconnected at the session handshake.

On incoming calls, if the password of the remote system does not match the password you specified here, this will be detected during the session handshake, recorded in the log file and the caller/callee will politely be shown the way out (i.e. disconnected).

If Nodelist support has been enabled by setting the NODELIST parameter, passwords will be fetched from there, unless the other system is not found in the nodelist.

Example:

```
Password "secret"
```

QUIT

This tells TrapDoor to exit (when waiting in answer mode) and is exactly the same as "ABORT F" or pressing Alt-X or Alt-Q.

(+) ARexx only command (@) asynchronous execution possible

Examples:

```
Quit  
@Quit
```

(NO)QUIET

If enabled, TrapDoor will run quietly in the background without opening any windows, screens etc. The logfile will still be written and you can still send ARexx commands to a TrapDoor running in QUIET mode.

Examples:

```
NoQuiet  
Quiet
```

REDIALDELAY seconds

This specifies the amount of time TrapDoor spends idling between calls to a system. Note that TrapDoor will not accept incoming calls while in "redial" state, so this option is only suited for points and will only work if you start TrapDoor with "TrapDoor Call ..." (i.e. you do not start it in answer mode).

Example: RedialDelay 120

RETRIES count

When calling a system, this instructs TrapDoor not to exit on a failed call (eg. line is busy), but to try it thus often again. Between the calls, TrapDoor will spend REDIALDELAY seconds waiting. Note that TrapDoor will not answer incoming calls while "redialdelaying".

Example:

Retries 5

REXXNAME portname

Using this configuration keyword, you can select the ARexx port name (the "host address" in ARexx terminology) of TrapDoor. If you run TrapDoor on multiple lines, be sure to set up a different REXXNAME for every invocation. TrapDoor will not start up if the port name is already in use.

Example:

RexxName "TrapDoor"

RINGS

Number of rings to wait before answering an incoming call. To turn off the answering feature, you can set this to any high-enough value (RINGS 5000 will probably never answer the phone).

Example:

Rings 1

RUN command-string

Causes TrapDoor to execute the given command asynchronously. If the command includes spaces, it must be enclosed in double-quotes; to include quotes within the quoted string, use the backslash ("\") as an escape character.

TrapDoor will not wait until the command returns, but merely continue to process ARexx messages and answer incoming calls. This is the big difference to the SPAWN keyword. In fact, "Run XXX" is exactly the same as "SPAWN \"Run XXX\"".

(+) ARexx only command

Examples:

```
Run "ed mail:TrapDoor.cfg"
Run "echo \"This command was started from TrapDoor.\""
```

SCREENMODE mode

This allows you to specify the screen where TrapDoor opens its windows. There are four possible modes:

WORKBENCH — TrapDoor will open its windows on the workbench screen.

CUSTOM — TrapDoor will open its own screen and place the windows there.

TRAPDOOR — Similar to **CUSTOMSCREEN**, TrapDoor will open its windows on its own screen. However, if you have multiple TrapDoors running at the same time, all invocations of TrapDoor will share the same "TrapDoor" screen. The screen will only close when the last copy of TrapDoor quits running.

ACTIVE — TrapDoor will use the screen with the currently active window and open its windows there. Note that this is rather dangerous as TrapDoor cannot control when the other program will close the screen. If this happens, TrapDoor will try to write to a screen that no longer exists and this will in most cases immediately crash the machine.

Examples:

```
ScreenMode WorkBench
ScreenMode TrapDoor
```

SERIALFLAGS serial-flags

If you use some other device than "serial.device", you might need to change this, too. Consult the documentation that came with your other device or use zero as default.

Example:

```
SerialFlags 0
```

SERIALNAME serial-device-name

If you happen to have a modem connected to some other device than the standard Amiga "serial.device", you can use this parameter to set up the correct device name. Usually, "serial.device" will just be about perfect. For Supra 2400zi modems, use **SERIALNAME "modem0.device"**.

Example:

```
SerialName "serial.device"
```

SERIALUNIT serial-unit-number

If your modem is connected to some other unit number than zero (on the device you set with **SERIALNAME**), change this appropriately.

Example:

```
SerialUnit 0
```

(NO)SHARED

(Don't) Open the serial device in shared mode. If you are running TrapDoor in conjunction with a BBS program or similar, you need to switch the serial device to **SHARED** mode, so that both programs can have it open at the same time.

Examples:

```
NoShared
Shared
```

(NO)SHOWREXX

In **SHOWREXX** mode, TrapDoor will display all ARExx commands that it processes in the status window.

You can toggle showrexx mode from the keyboard using Alt-S.

Examples:

```
NoShowRexx
ShowRexx
```

SLOWMODEM modem-command-delay

Some modems won't understand incoming data at full speed while in command mode. Others take some time after a reset (ATZ or DTR dropped) until they will react to incoming commands again. If your modem doesn't seem to understand the commands TrapDoor is trying to send, feel free to change (increase) this parameter. Also see chapter "The Modem" for some suggested values.

SLOWMODEM changes a number of timings in TrapDoor, the most important being: (a) the time that TrapDoor waits between lowering and raising of DTR when it tries to reset the modem or hang up,

(b) the time that TrapDoor waits when it encounters a tilde ("~") character in a modem command string (MODEMINIT, MODEMDIAL, MODEMANSWER).

Example:

```
SlowModem 7
```

SPAWN command-string

Causes TrapDoor to execute the given command. If the command includes spaces, it must be enclosed in double-quotes; to include quotes within the quoted string, use the backslash ("\") as an escape character.

TrapDoor will wait until the command returns. It will not execute any other ARexx commands, nor will it answer incoming calls while the command is executing.

(+) ARexx only command

Examples:

```
Spawn "ed mail:TrapDoor.cfg"
Spawn "echo \"Spawned from TrapDoor!\""
```

STATUS what

Depending on the argument, this command will return various information about the state TrapDoor is in, the result of previous calls etc.

The following status specifiers are recognized:

C ... Returns the cost of the last call made.

D ... Reports the result string of the dial (or call), such as BUSY, NO DIALTONE.

S ... Reports the serial status of TrapDoor, which indicates whether TrapDoor is currently waiting for a call, answering a call or making an outgoing call.

Valid return values are:

IDLE: TrapDoor is idle, waiting for a call

OUTGOING: TrapDoor is making an outgoing call

INCOMING: TrapDoor is answering an incoming call

X ... Queries the status of the last transfer. It returns an integer number, where:

0: everything okay

2: transmission error (too many retries)

3: end-of-transmission

4: file skipped

5: user break (Ctrl-C or ESC)

6: carrier lost

7: disk i/o error
8: remote canceled transmission
9: internal error

(+) ARexx only command

(@) asynchronous execution possible

Examples:

```
Status D
@Status S
```

STATWINDOW window-specification

Similar to LOGWINDOW, this changes the position and size of the status window.

Example:

```
StatWindow 30/155/580/37
```

(NO)SWEPUULSE

Applies a special number translation to the dial string before sending it to the modem. Special feature for swedish people with non-swedish modems.

Examples:

```
NoSwePulse
SwePulse
```

SYSOP sysop's-name

Sysop's name, will be sent to the other system during session negotiation. The length of this string is limited to 20 characters.

Example:

```
Sysop "Maximilian Hantsch"
```

TASKPRI priority

Use this to select the AmigaDOS/Exec task priority of TrapDoor. Normal tasks operate at priority 0. It is often advisable to set the priority of TrapDoor to 1 or 2, so that mail sessions are not slowed down by other activity such as mail importing/exporting.

If the keyword is not used, TrapDoor will run at the priority of the invoking process.

Example:

```
TaskPri 1
```

TESTFREQ

To be able to test your file request server programs and scripts, call TrapDoor with "TrapDoor testfreq". This will cause TrapDoor to look for a File.Request file in the inbound directory and, if one is found, call the file request server (specified with the FREQUEST keyword). The result of the file request server should be a .RLO file in your outbound directory, which you can check to see whether everything works all right.

Please note that the request file should be called "File.Request".

Example:

TestFreq

(NO)WAZOO

Enable (disable) WaZoo mode. With WaZoo disabled, TrapDoor will only attempt to connect using the FTS-0001 (Lotek) protocol. No YooHoo will be done.

Examples:

```
NoWaZoo
WaZoo
```

(NO)WRAPLINES

With WrapLines enabled, TrapDoor will try to wrap lines that are sent to the log window and continue them in the next line, if they are too long to be displayed on a single line.

The logfile will be unaffected by the setting of this switch. Only the output in the window is reformatted.

Examples:

```
NoWrapLines
WrapLines
```

(NO)ZEDZAP

Enable (disable) ZedZap mode. ZedZap is a slight variant of the ZModem transfer protocol — on large files, this is one of the fastest protocols around. ZedZap will automatically switch block sizes depending on modem speed and quality of the line. It will also resume an interrupted transfer if possible.

ZedZap only works in WaZoo or EMSI mode and only if the other side also supports it.

Examples:

```
NoZedZap
ZedZap
```

TrapDoor with ARexx

The following section outlines Trapdoor's ARexx support. In a normal DLG installation, you will not have to communicate with Trapdoor directly as DLGMail does this automatically. This section, however, is provided in the manual for completeness.

TrapDoor includes an ARexx port — while in answer mode waiting for incoming calls TrapDoor also accepts ARexx messages. The port name, which is also called the "host name" in ARexx symbolics, is usually "TrapDoor", unless you change it with the REXXNAME keyword. Use this name in Rexx "Address ..." statements to select TrapDoor.

All commands and keywords listed above can be sent to the ARexx port and will be understood, although some might not behave as expected. (For example, if you change SERIALNAME while in answer mode, TrapDoor will not switch to the new device. Instead, you should terminate TrapDoor and run it again with a different setup.)

Apart from that, almost everything that can be set up from the command line (or a config file) can also be done via the ARexx port.

Some commands shown here can only be used from ARexx (or TrapTell). Those keywords are marked with

(+) ARexx only command

Attempts to use them from CLI will cause nothing but an error message.

Some ARexx commands can also be sent for immediate execution. In that case, the command will be executed immediately, no matter whether or not TrapDoor is just connected to another system and a mail session takes place. For immediate execution, the command must be prepended with an "at" character ("@"). Such keywords are marked with

(@) asynchronous execution possible

All other commands sent via ARexx will be executed whenever TrapDoor is idle and waiting for a call.

Here is an example:

```
/* This is a Rexx script to call my boss. TrapDoor must already be running in
Answer mode. */
address "TrapDoor"
"Call Boss"
"Status D"
say "Modem returned" RESULT "from the last call."
"Status X"
say "The last call terminated with error" RESULT"."
```

TrapDoor without ARexx

Should you happen not to have a copy of ARexx handy, no problem — the tool TrapTell simulates an ARexx server, sending a message to TrapDoor and waiting for the results.

Here is an example. It stops a running TrapDoor by sending it an "Abort F" command.

```
TrapTell "Abort F"
```

First steps

Here's an easy example of how to use ARexx (or TrapTell). First, let's start up the mailer in answer mode, but not react to incoming calls:

```
TrapDoor answer rings 50000
```

Then, I'd like to call node 1:200/300 and get the connect result string:

```
TrapTell "call 1:200/300" TrapTell "status d"
```

Try it, you should soon become familiar with that method of controlling programs. There are some example AmigaDOS scripts that use the TrapTell command in the Scripts/ subdirectory on the distribution disk (or in the distribution archive) and there are some more ARexx programs that make use of TrapDoor's ARexx port in the rexx/ subdirectory.

You, too, can have a receding hairline, suffer a nervous breakdown, and precipitate disputes with your spouse. Here is all you need.

Example Setup

```
; CONFIGURATION FILE FOR TRAPDOOR V1.80 - MODIFICATIONS WILL NEED TO BE
; MADE FOR EARLIER VERSIONS OF TRAPDOOR AS NOT ALL THE KEYWORDS WILL
; BE RECOGNIZED...
;
; things you need to customize
;
NODE 1:114/52.0      ; your MAIN network address
BOSS 1:114/52.0      ; for node use, set this to your own address
```

```

AKA "1:5252/0.0 1:5252/1.0" ; if you set up a pointnet, include the
                             ; pointnet in the list of AKA's. Add AKA's
                             ; as necessary for other networks you may
                             ; belong to
NAME "Your BBS Name Goes Here"
SYSOP "Your Name"
BANNER "What the user sees when he connects to TrapDoor"
BBSCHAR '.'
; serial things you may need to change
SERIALNAME "serial.device" SERIALUNIT 0 SERIALFLAGS 0
BAUD 19200
LOCK
; use these only if you have an error-correcting modem (HST, HST Dual Std,
; V.32, MNP. Adjust the ARQSTRING parameter as required.
SNIFFARQ|
ARQSTRING "ARQ"
; Pick your poison for having TrapDoor load the BBS. The commented out
; pair are for memory hungry systems and tends to be a wee bit less
; reliable. The pair in use below leaves TrapDoor in all the time and
; allows it to come back for the next call instantly upon a user
; disconnect. There is a setting in the ZMAIL.CFG file that needs to
; be altered if you change this mode
BBSMODE SPAWN
BBSCOMMAND "DLG:immed TRO %b %B -w"
; OR
;BBSMODE EXIT
;BBSCOMMAND "DLG:immed TRO %b %B"
; serial things you probably won't need to change
; unless you don't have an HST. You need to set your
; modem up ahead of time with all the proper S-register
; settings and other assorted goodies - save them to
; NVRAM so that an ATZ will bring them all up as default
; HST Dual Standard users may want to alter the MODEMDIALPRE
; to send an altered ATB setting. See your dox.
MODEMINIT ""ATZ|"
MODEMDIALPRE ""ATDT "
MODEMDIALPOST "|
MODEMANSWER "ATEQA|"
MODEMHANGUP "|^"
7WIRE
SHARED
MINBAUD 1200
NOMAXBAUD
DSR
SLOWMODEM 8
ADJUST 0
; things you probably shouldn't change unless you have specific problems
; connecting to specific systems...
DIRECTZAP
ZEDZIP
WAZOO

```

```

DIETIFNA
ZEDZAP

; EMSI is not always beneficial, depending on who you connect with. Try
; using EMSI and if you have trouble, disable it with NOEMSI
EMSI
; Other miscellaneous things to leave alone...

ANSWER
RINGS 1
BBSINOUT ""
REXXNAME "TrapDoor" ; This setting is for your MAIN Trapdoor line
OUTBOUND "Outbound:" ; LEAVE ALONE!
INBOUND "Inbound:" ; LEAVE ALONE!
NODELIST "NodeList:" ; LEAVE ALONE!
RETRIES 0 ; no affect when operated as a node
REDIALDELAY 0 ; no affect when operated as a node
TASKPRI 0 ; more than adequate for most systems. if
; you do a lot of multitasking in the
; background, bumping this to 1 may help

; Make it look like you want it to...
LOGWINDOW 0/11/640/100
STATWINDOW 0/11/640/89
SCREENMODE CUSTOM
BACKGROUND
NOAUTOOVERSCAN
NOINTERLACE
SHOWREXX

; This is for after a mail session, not after a BBS call... Leave this
; alone
AFTERSSESSION "run FIDO:DMC process"

; Will you allow FREQuing of files from your system? If so, enable it
; here. Substitute your own files list name for AC_FILES.TXT.
FREQUEST "DLG:TPTFreq MAIL:AC_FILES.TXT %i %o"

; call accounting - see the manual!
ACCTMAX "-1 -1 3 -1 -1 2"
ACCOUNTING

; logging goodies
LOGFILE "Logs:TrapDoor.LOG"
LOGLEVEL 0:2 ; debugger
LOGLEVEL 1:2 ; link
LOGLEVEL 2:2 ; transfer
LOGLEVEL 3:2 ; system
LOGLEVEL 4:2 ; modem
LOGLEVEL 5:2 ; session
LOGLEVEL 6:7 ; security
LOGLEVEL 7:2 ; outside
LOGLEVEL 8:5 ; information
LOGLEVEL 9:2 ; scheduler

; refer to the TD documentation for setting of function keys and other
; things...

```


Extensions

The file extensions tell TrapDoor how to treat a certain file.

Packet files

These files contain packed mail (Note: not compressed (ARC)mail). This is often used for matrix mail, as these packets are easily built and matrix mail normally doesn't get so large that it needs compression. They are sent 'as is' to the other system. During the transfer, the name of such a file is changed to "abcdefgh.PKT", where "abcdefgh" is a unique 8-digits hexadecimal number (in fact, a timestamp).

#?.OUT ... Normal, meaning that this packet hasn't been processed further. If left unprocessed, it will be treated the same as a .DUT packet. #?.HUT ... Hold this packet for pickup by the remote system. #?.CUT ... The other system can receive Continuous Mail. #?.DUT ... Direct, meaning the other system can NOT receive Continuous Mail.

Flow files

Files are also sent through FidoNet. File attach files tell TrapDoor what files to send (or hold) for whom. File attach files are also called 'flow files' after the .FLO file extension.

Flow files store the path and name of files that should be sent to the other system. Each line in the flow file refers to one file. Additionally, there may be one of the following special characters at the first position in the line, indicating that the file needs special processing after sending.

Valid special characters:

- # ... Truncate this file to zero length
- ^ ... Delete this file (with logging)
- ... Delete this file (without logging)
- ~ ... Don't send this file (has been sent previously)

Examples:

```
DH0:Files/outgoing/special/sendme.zoo
t:trapdoor.zoo
#MAIL:Outbound/FFEB0034.MO1
-t:delete.me
```

Flow file extensions are:

#?.FLO ... Normal, meaning that this flow file hasn't been processed further. If left unprocessed, it will be treated the same as a .DLO flow file.

#?.HLO ... Hold these files for pickup by the remote system.

#?.CLO ... The other system can receive Continuous Mail.

#?.DLO ... Direct, meaning the other system can NOT receive Continuous Mail.

Compressed Mail files

These files are not automatically sent. Their names must be listed in one of the #?.?LO files. Usually, the filename of these files follows a 2-dimensional naming method: The first 4 hex digits contain the difference between the net numbers of the originating and the destination system, the second 4 hex digits the difference between the node numbers.

The extension of compressed mail files is built of the first two digits of the name of a weekday, i.e. MO, TU, WE, TH, FR, SA or SU, plus one decimal digit to prevent duplicates, for example, "MO3" or "FR0".

TrapDoor supports 4-dimensional compressed mail bundles. These are named just like the other 4-dimensional files, with the same extension as their 2-dimensional counterpart, for example "2.310.6.4.TU3". The names of 4-dimensional compressed mail files must be listed in #?.?LO files, just as with their 2D counterpart.

Request files

These files are sent to the other system 'as is' for further processing. Each line in these files contain the name of a file you'd like to request from the other system and possibly, following an exclamation mark, a password for the file.

Example:

```
FILES
TRAPDOOR.ZOO
SECRET.ARC !ILBM
```

Examples

```
0136000b.out ... a normal mail packet for 310/11
01360003.clo ... a file attach file for 310/3, will be sent as continuous mail
0136000c.hlo ... a file attach file, held for pickup by 310/12
```

Or, using the new 4-dimensional naming scheme:

```
1.163.109.0.hut ... a normal mail packet, held for pickup by 1:163/109
2.310.6.5.dlo ..... a direct file attach file for 2:310/6.5
2.310.6.4.M02 ..... a compressed mail bundle for 2:310/6.4
2.246.3.0.req ..... a request file for 2:246/3
```

The Inbound Directory

The inbound directory stores all files received from other systems. TrapDoor does not associate inbound filenames with a certain meaning, this is left for the Scanner/Tosser software (i.e. TrapToss, ConfMail).

TrapDoor currently manages the inbound directory filenames & filenotes like this:

<filename> := <msdos_filename>

'!' <msdos_filename> ' ' <fido_adr>

<fido_adr> := <zone> ':' <net> ':' <node> ':' <point>

<zone> := integer

<net> := integer

<node> := integer

<point> := integer

<filenote> := <field> <field> ...

<field> := <tag> ' ' <contents> ';' ;

<tag> := anything_except_space

<contents> := anything_except_semicolon

Currently defined tags are:

FileName

From

Length

Secure

Trx

When a file comes in, it is received under a temporary filename. When the file has been received successfully, it is renamed to the final filename. Should a file with this name already exist, TrapDoor does the usual filename bumping (see "Bumping Filenames"), but stores the original filename in the filenote in a "FileName" field. Additionally, each inbound file is tagged with a "From" field in the filenote.

When a file transfer fails (carrier lost etc.), the temporary file is renamed to "! filename zz.nnn.ooo.pp", where <filename> is the original filename, and <zz.nnn.ooo.pp> is the FidoNet address of the sender. These files are also tagged with a "FileName" field in the FileNote. Additionally, TrapDoor adds a "Length" field to the FileNote, which specifies how long the file should be.

When TrapDoor begins to receive a new file, it tries to find a "! filename zz.nnn.ooo.pp" file that matches the incoming file. If such a file exists, and the filesize compare successfully (remember: "Length" tag in the filenote!), TrapDoor will resume transfer (ZedZap sessions) at the end of the file. FTS-1 and Dietlfna can't resume transfer, but you can abort such a file transfer and lateron resume with ZedZap.

When TrapDoor resumes receiving a file, it will first rename it to a temporary filename again and clear the filenote. Further actions are taken as described above.

The "Secure" tag stores the security measures under which a file was received. Files received from systems listed in the nodelist get the "NL" flag, and files received during password-protected sessions are marked "PW". Both flags can be set if a file was received from a listed system in a password-protection session, in which case they will be separated by a comma (",").

The "Trx" tag is used to store the Transaction ID of the session in which the file was received. Every session, or "transaction" with another system is associated with a unique identifier, the "transaction id". This identifier is also listed in the logfile as "TrxID". A transaction id consists of one single hexadecimal number, plus possibly a slash and another hexadecimal number giving the transaction id which the remote system associated with this particular transaction. The latter is only shown if the other end sent us their transaction id, which can only happen in an EMSI handshake.

Please note that the order of the filenote fields is insignificant.

Examples:

Normal inbound file, received from a known node with password-protection:

```
ffeb0034.mo1
: From 2:310/3; Secure NL,PW; Trx 27b23ba7;
```

Bumped inbound file:

```
ffeb0034.mo2
: From 2:310/3; FileName ffeb0034.mo1; Trx 27c138f7/27c138fb;
```

Aborted transfer:

```
! ffeb0034.mo1 2.310.3.0
: FileName ffeb0034.mo1; Length 23862; Trx 27c139ae;
```

This format should be rather flexible if future extensions have to be added (limit is 80 chars for all the filenote fields), but still remains totally compatible with ConfMail & other utilities that assume MS-Dos style filenames.

Bumping Filenames

TrapDoor does not overwrite inbound files. Instead, the incoming filename is "bumped", i.e. a unique name is created by cautiously modifying the original filename until a new, unique name is found.

Bumping proceeds like this:

(a) If the filename matches "#?.?UT", it is renamed to a random "#?.PKT" file. This is necessary, as some MS-Dos software does not rename .CUT files into .PKT when transmitting them with the SeaLink protocol. Therefore, TrapDoor tries to correct this mistake. After this step, bumping continues at (b). [00bf13f4.OUT -> 27bc34f5.PKT]

(b) If the filename matches "#?.PKT", it is given some random hexadecimal number plus the ".PKT" extension again. Thus, packet files stay packet files, even when bumped. Only if a unique name cannot be found within ten tries, method (d) is used for renaming the file. [27bc34f5.PKT -> 28c39d7f.PKT]

(c) If the filename matches "#?.(M0ITUIWE|THIFRISAI|SU)([0-9])", the very last digit of the filename is modified. If a unique name cannot be found by changing the last digit, method (d) is used. [00bf3f55.mo0 -> 00bf3f55.mo1]

(d) If the filename is not unique, TrapDoor will append a comma (",") and a number. The number will be incremented as long as the name is not unique. [test.txt -> test.txt,1 -> test.txt,2 -> usw.]

(e) continue at (d) until a unique name is found.

Caveats

The only problems we have encountered are usually caused by wrong or incomplete RS232 cabling, modem settings or slow modems, and last, but not least, people not reading the documentation. Be sure to especially examine the chapter about the modem carefully if you encounter problems.

Note that if you specify a nodenumber in the CALL statement, the nodelist search will be done after the commandline has been completely parsed. This means that if you execute for example

```
TrapDoor CALL 2:310/6 PASSWORD secret
```

the password will be taken from the nodelist and not from the commandline. This is probably not what you expected. We hope to fix this in some future version. Meanwhile, change the password for that node with the "setpasswd" utility.

TrapDoor usually runs with the standard 4000 bytes stack size. Should you experience strange hangups, crashes or Guru Meditations, try to raise the stack size with the CLI command "Stack" before running TrapDoor.

License

1. This license applies to the product called "TrapDoor", a set of programs for the Amiga computer, published by Maximilian Hantsch and Martin Laubach under the concepts of ShareWare, and the accompanying documentation, example files and anything else that comes with the original distribution. The terms "Programs" and "TrapDoor" below, refer to this product. The licensee is addressed as "you".

2. You may not disassemble, decompile, re-source or otherwise reverse engineer the programs.

Disclaimer

No warranty, either express or implied, is made with respect to the fitness or merchantability of TrapDoor.

Maximilian Hantsch and Martin Laubach (referred to as "the authors"), reserve the right to not develop any future versions of TrapDoor.

The authors will try to make a good faith attempt at correcting any problems if any are discovered, but are in no way required, nor bound to correct them.

The authors neither assume nor accept any responsibility for the use or misuse of these programs. They also will not be held liable for damages or any compensation beyond the original registration fee due to loss of profit or any other damages arising out of the use, or inability to use these programs.

Neither Maximilian Hantsch nor Martin Laubach will be liable for any damage arising from the failure of these programs to perform as described, or any destruction of other programs or data residing on a system attempting to run the programs. While we know of no damaging errors, the user of these programs uses it at his or her own risk.

Acknowledgements

The following names are either trademarks or the efforts of the person and/or company listed:

Amiga and AmigaDOS are trademarks of Commodore-Amiga, Inc.

ARexx by William S. Hawes, Wishful Thinking Development Corp.

BinkleyTerm Amiga by Jergen Hermann.

BinkleyTerm by Alan D. Applegate, Robert Hartman and Vincent Perriello.

Chameleon Editor by Jergen Hermann.

ConfMail by Bob Hartman, Spark Software Inc.

D'Bridge by Chris Irwin.

DLG Professional by TelePro Technologies.

EchoMail by Jeff Rush.

EMSI by the EMSC.

FFSA by Carl-Christian Kanne.

Fido and FidoNet are trademarks of Tom Jennings, Fido Software.

FrontDoor by Joaquim Homrighausen.

Inspiration by US Robotics' Dual HST.

License Agreement inspired by Jack Radigan and the GNU General Public License.

Many Thanks to Alexander Holy, Andreas Levenitschnig,

Andreas Dolleschal, Arnout Grootveld, Cafe Heine,

Felix Kasza, Heiko Richter, Johannes Mistelbauer,

Jergen Otte, Manfred Schedler, Many Others,

Michael D. D'Bridge, Steve Lewis,

Tony Miller and Werner Schlagnitweit.

Moral Support by Martina Patzelt and Snoopy.

MS-DOS is a trademark of Microsoft Corporation.

Opus by Wynn Wagner III.

Paragon by Jon Radoff.

Proofreading of this manual by Arnout Grootveld.

SEALink, SEAdog are trademarks of System Enhancement Associates.

Special Thanks to all our Registered Users.

TrapDog by Peter Wlcek.

TrapDoor, TrapList, TrapPoll, TrapTell and traplist.library
by Maximilian Hantsch and Martin Laubach.

TrapToss by Rene Hexel.

YooHoo by Wynn Wagner III.

ZModem by Chuck Forsberg.

Appendix

This appendix contains lists of reference information. This information is available elsewhere in the manual, but has been collected here for ease of use.

%Switches

A %switch is a percent character followed immediately by one of a given set of keywords that DLG will recognize. %Switches can be used in DLG text files, menu sets, and language files.

When DLG encounters a %switch in a text file as it is being displayed, it will substitute a piece of information for the %switch. There are a number of different switches that DLG provides for, which makes for some very interesting, and personalized text files. Case is not important to %switches. Switches that refer to user data will always report the data from the current user's account.

%Switches can be optionally followed by a period and a number. This will cause DLG to format the substituted text to be exactly as wide as the indicated number. Substitutions that are longer than the switches are cut off, while shorter substitutions are padded out with spaces. This makes it easy to create attractive formatting with variable information.

Here is a complete list of all DLG %switches with an explanation of each one:

User Personal Information Switches:

%NAME	-The full name of the user
%FIRST	-The user's first name
%LAST	-The user's last name
%UNAME	-The user's name with an underscore character (i.e. Kim_Green)
%ADDRESS	-The address of the user
%CITY	-The user's city
%PROVINCE	-The user's State or Province
%COUNTRY	-The user's Country
%POSTAL	-The user's ZIP or Postal Code
%PHONE	-The user's phone number
%BYEAR	-The year of the user's birthday
%BMONTH	-The month of the user's birthday
%BDAY	-The calendar day of the user's birthday
%AGE	-The age of the user

User System Information Switches:

%JOINED	-The date the user joined the system
%LASTCALL	-The date the user last called the system
%CALLS	-The number of calls the user has made to the system
%ALIAS	-The user's alias
%PASSWORD	-The user's password
%COMPUTER	-The type of computer the user has
%LEVEL	-The user's User Level
%SCLNGTH	-The length of the user's screen
%SCWIDTH	-The width of the user's screen
%HELPLVL	-The user's help level - either Novice, Intermediate, or Expert
%PROTOCOL	-The user's chosen download protocol
%UPROTO	-The user's chosen upload protocol
%SYSPAGES	-The number of times the user has paged the SysOp
%DAYLIMIT	-The user's daily time limit in minutes
%TLTODAY	-The number of minutes left in the user's daily limit

%SESLIMIT	-The user's time limit per session
%TLCALL	-The number of minutes left in the user's current session
%TUTODAY	-The number of minutes the user has used today
%TUTOT	-The number of minutes total that the user has been on the system
%DIRLIMIT	-The size of the user's allowed private directory
%MSGENTER	-The number of messages the user has written
%MSGREAD	-The number of messages the user has read
%BYTESUP	-The number of bytes the user has uploaded
%BYTESDN	-The number of bytes the user has downloaded
%FILESUP	-The number of files the user has uploaded
%FILESDN	-The number of files the user has downloaded
%DLBYTES	-The number of bytes the user is allowed to download (-1 if no limit)
%RATIO	-The user's upload/download ratio limit
%LASTMSG	-The number of the last message area the user visited
%LASTFILE	-The number of the last file area the user visited
%PORT	-The three letter name of the port the user is current logged into
%BAUD	-The baud rate of the user's current connection
%CREDIT	-The user's current NetMail credit
%NETPRIV	-The state of the user's NetMail privileges
%UUCP	-The user's UUCP status
%ANSI	-The user's ANSI status (either "color" or "mono")
%MENUSET	-The user's selected menu set

General Information Switches:

%DATE	-The current date and time
-------	----------------------------

Display Control Switches:

(NOTE: unlike the variable %switches above, these %switches MUST appear at the START of a line. If there is a numeric parameter required, be sure to follow the parameter with at least one space.)

%MOREOFF	-Disable "More" prompts
%MOREON	-Restore "More" prompts to user's settings
%WRAPOFF	-Disable word-wrap
%WRAPON	-Restore word-wrap
%POSOFF	-Disable ANSI screen-positioning
%POSON	-Restore ANSI screen-positioning to user's settings
%CLROFF	-Disable screen clears
%CLRON	-Restore screen clears to user's settings
%COLOUROFF	-Disable ANSI colour
%COLOURON	-Restore ANSI colour to user's settings
%RETURN	-Display "Press RETURN" prompt
%DOMORE	-Force a "More" prompt
%INDENT.<x>	-Indent text by <x> number of spaces
%SETWIDTH.<x>	-Set the user's screen width to <x> for current file
%PAUSE.<x>	-Pause for <x> seconds
%SLOW.<x>	-Insert <x> number of ticks (50th of a second) between characters
%BREAKON	-Enable CTRL-C breaking
%BREAKOFF	-Disable CTRL-C breaking

ANSI Control Switches

%a<n>	-Change the following text to colour <n> (0 to 7)
%ab	-Change the following text to bold
%ai	-Change the following text to italic
%au	-Change the following text to underlined

%ar	-Change the following text to reverse video
%af	-Cause the following text to blink (not supported in most Amiga terminal programs)
%ao	-Reset all text modes to normal

Built-In Functions

The following is a list of the built-in functions for all DLG modules:

MESS:

DisplayMenu	- This function is available in all DLG menus. It displays the current menu.
Help	- This function is available in all DLG menus. It displays the current menu and prompts the user to choose a command letter. The help text associated with that menu item is then displayed.
MSG_ChangeArea	- This function will allow a user to select a new message area. If selected without an accompanying area number (as in "A20") then the command will prompt the user to enter an area number, or optionally display a list of all available message areas, and then prompt again for an area number.
MSG_ContRead	- This function will enable the "continuous read" mode of the message base. The user will be prompted to provide settings for ANSI stripping, and more prompts.
MSG_Correct	- This function will enable users to edit messages they have just read.
MSG_DeleteAll	- This function will enable users to delete all mail from their private message areas. This function will only be active if users are IN their private message area at the time it is invoked.
MSG_EditSig	- This function enables users to edit their signature files. Users will be presented with a list of possible signatures to edit - local, UseNet, EchoMail.
MSG_Forward	- This function will allow a user to move or copy the message just read to a different message area, or to another user.
MSG_FwdRead	- This function allows the user to select the forward reading mode (the default).
MSG_Kill	- This function allows the user to delete the message just read.
MSG_LexCheck	- This function allows the user to "lex check" the message just read.
MSG_ListReaders	- This function lists the names of the users who are regular readers of the current message area.
MSG_NewScan	- This function will scan the message areas listed in the user's global scan preferences, looking for new messages.
MSG_PvtArea	- This command will switch a user to their private message area.
MSG_ReadNext	- This function will allow the user to read the next message in the area. If there are no more messages in the current area, this command will take the user to the next available message area from their global message area list. The "next message" is dependent upon the reading direction, and reading mode chosen (tag, thread, or normal).
MSG_ReadOrig	- This function will allow the user to read a message that the message just read is a reply to.

MSG_ReadReply	- This function will allow the user to read a message that is a reply to the message just read.
MSG_ReadTagged	- This function will allow the user to read messages that they have tagged for future reading, or have been tagged for them by the BBS when they were entered into the system.
MSG_Reply	- This function will allow the user to reply to a message that they have just read.
MSG_ReRead	- This function will allow the user to re-read a message that they have just read.
MSG_RevRead	- This function will allow the user to read messages in reverse order.
MSG_Search	- This function will activate the header search feature. The user will be prompted for a search string if one was not provided with the command. If the search string is empty then the search feature is disabled.
MSG_SelectSig	- This function will allow a user to select a different SIG. If no accompanying SIG number is specified (as in "S1") then the command will prompt the user to enter a SIG number, or optionally display a list of all available SIGS, and then prompt again for a SIG number.
MSG_SkipThread	- When in thread read mode this command will allow the user to skip threads they are not interested in. All messages remaining in the current thread will be skipped by the message reader.
MSG_TagRead	- This command activates a header scan mode where only the headers of messages are displayed. Readers can then tag messages that they want to read in full in tag read mode.
MSG_ToggleThread	- This function toggles thread reading mode on and off. In thread reading mode, all messages that have the same subject line are considered to be part of a "thread", and will display one after another. In normal reading mode, messages are read in the same order that they arrived in the message area. Topics are interwoven amongst each other, and conversations can be more difficult to follow in normal reading mode. When in thread reading mode you have the option of skipping message threads you are not interested in reading.
MSG_UpdatePtr	- This function will update the reader's high message pointer to the highest message in the area. This provides new users with a quick way of "catching up" to the most current messages on your system.
MSG_Write	- This function enables users to compose messages, with the use of the editor of their choice (determined in their user options).
MSG_WriteBlt	- This function enables users to compose bulletins, with the use of the editor of their choice. Bulletins differ from messages in that they appear automatically to all users as they log into the system, and that they have an "expiry date", after which they are removed from the system.
FILE:	
DisplayMenu	- This function is available in all DLG menus. It displays the current menu.
Help	- This function is available in all DLG menus. It displays the current menu and prompts the user to choose a command letter. The help text associated with that menu item is then displayed.

File_ChangeArea	- This function will allow a user to select a new file area. If selected without an accompanying area number (as in "A20") then the command will prompt the user to enter an area number, or optionally display a list of all available file areas, and then prompt again for an area number.
File_Comment	- This function will allow a user to add a comment to the file description just read, using the editor they have chosen in their user options.
File_Download	- This function will allow a user to download a file. If the user has just read a file description, the option will be to either download that file or files that the user has tagged for batch downloading. If the user has not just read a file description, then the system will assume they want to download the tagged files, or if there are no tagged files, prompt the user for the name or number of a file to download from the current file area. If the user has a preset download file transfer protocol, then that will be used automatically. If the user has no preset protocol, then DLG will prompt the user to select one from a list of all protocols available.
File_Edit	- This function will allow the user to edit the file just read. Edited items include the file description, file name, and file size.
File_EditSig	- This function enables the user to edit their signature files.
File_GlobalList	- This function lists all of the files in all of the file areas that the user has selected in their user options. The user will be asked for listing options - forward, reverse, alphabetical forward, alphabetical reverse, all files, new files, since last call, since # of days, since date, in a given range of dates, search for a specific filename, or by filename and description.
File_Kill	- This function will allow a user to delete the file and description just read.
File_List	- This function is very similar to File_GlobalList, except that it functions only in the current file area, not all files that the user has selected to scan.
File_ListBatch	- This function allows the user to list all of the files that they have tagged for batch downloading.
File_NewScan	- This function will search all of the file areas that the user has selected in their global file area list in the user options, for new files. A new file is one that the user has not "seen" yet, not new files since last log-in.
File_PvtArea	- This function will take the user to their private file area.
File_Read	- This function will enable the user to read each new file description in an area. On the default DLG system, this is the command implied when the user presses RETURN at any prompt. If a number is supplied, then the file with that number is read instead of the next available file.
File_RemoveBatch	- This function will enable the user to delete some or all of the files that they have tagged for batch download.
File_SelectSig	- This function will allow a user to select a different SIG. If no accompanying SIG number is specified (as in "S1") then the command will prompt the user to enter a SIG number, or optionally display a list of all available SIGs, and then prompt again for a SIG number.
File_Tag	- This function adds the file just read to the user's list of files for batch download. If the user has not read a file description, they will be prompted for the name or number of the file to tag.

- File_Transfer - This function enables the user to transfer the file just read to another file area or to a user's private file area.
- File_Upload - This function enables the user to upload a file to the current file area, or to a user's private file area. If the user has an upload protocol preference selected in their user options, it will be used automatically. If not, the user will be prompted to select a file transfer protocol from a list of all available protocols.
- File_Validate - This function enables the user to validate an uploaded file. The validated file is moved from the designated validation file area to the area it was originally uploaded to, or to a file area of the user's choice.
- File_ViewArchive - This function enables the user to list the files contained within the current file, if it is one of the configured archive file types.
- File_ViewNext - This function enables the user to see the description of the next file in the current file area, or to go to the next available file area from their list of global file areas.

CONFUSER

- CONFU_CreateRoom - This function allows you to create a conference room. Prompts will be given to guide the creation process.
- CONFU_EnterRoom - This function allows you to enter a conference room.
- CONFU_Exit - This function exits the module.
- CONFU_ListRooms - This function lists all available conference rooms.
- CONFU_ListUsers - This function will list all users in a given conference room.

FILEAREA

- FILEA_Add - This function allows you to create a file area.
- FILEA_Delete - This function allows you to delete a file area.
- FILEA_Edit - This function allows you to edit the characteristics of a file area.
- FILEA_Exit - This function will exit the module.
- FILEA_List - This function will list all file areas.

FILEMAINT

- FILEM_BatchUp - This function performs a batch upload operation. You will be prompted to provide a suitable description for each file.
- FILEM_ChangeArea - This function allows you to choose which area to work on.
- FILEM_EditFile - This function allows you to edit the characteristics of a file — its header, size, description, and FREE status.
- FILEM_Exit - This function allows you to exit the module.
- FILEM_Freshen - This function performs a freshen on the file area, making sure that its files match up with the descriptions, and maintaining the necessary index structure.
- FILEM_GlobalEdit - This function allows you to globally edit the FREE status of every file in an area.
- FILEM_ListFiles - This function allows you to list the files in an area.

FILEM_TurboUp	- This function allows you to upload a number of files using a generic description for all of the files, or using each file's filenotes to provide the description.
---------------	---

FILEUSERS

FILEU_Add	- This function allows you to add a user to a file area.
FILEU_AreaCopy	- This function allows you to copy the user list from one file area to another.
FILEU_Delete	- This function allows you to remove a user from a file area.
FILEU_Edit	- This function allows you to edit the access of a user in a file area.
FILEU_Exit	- This function exits the FileUsers module.
FILEU_List	- This function allows you to list all users in a file area.
FILEU_ListArea	- This function allows you to list all areas available to a particular user.

GROUP

GROUP_Add	- This function allows you to create a new group account. A group account consists of a name for the account, the name of a designated GroupOp, and the members of that group.
GROUP_AddUser	- This function allows you to add a user's name to an existing group.
GROUP_Del	- This function allows you to remove a group.
GROUP_DelUser	- This function allows you to remove a user from a group.
GROUP_Exit	- This function exits the Group module.
GROUP_List	- This function will list all available groups.
GROUP_ListUsers	- This function will list all the users in a group.

MSGAREAS

MSGA_Add	- This function will allow you to create and define a new message area.
MSGA_Delete	- This function will allow you to delete an existing message area.
MSGA_Edit	- This function will allow you to edit the characteristics of an existing message area.
MSGA_Exit	- This function will exit the MSGAreas module.
MSGA_List	- This function will list all available message areas.

MSGUSERS

MSGU_Add	- This function will allow you to add a user to message area
MSGU_AreaCopy	- This function allows you to copy the user list from one message area to another.
MSGU_Delete	- This function allows you to delete a user from a message area.
MSGU_Edit	- This function allows you edit the access of a user in a message area.
MSGU_Exit	- This function exits the MSGUsers module.
MSGU_List	- This function lists all users in a given message area.
MSGU_ListArea	- This function lists all areas available to a given user.

PORT

- PORT_AddCompTypes - This function allows you to add new entries to the list of computer types.
- PORT_AddDisplay - This function allows you to create and define a new display configuration file.
- PORT_AddGlbSet - This function allows you to create and define a new global configuration file.
- PORT_AddModem - This function allows you to create and define a new modem configuration file.
- PORT_AddPort - This function allows you to create and define a new port configuration.
- PORT_DelDisplay - This function allows you to delete an existing display configuration.
- PORT_DelGlbSet - This function allows you to delete an existing global configuration.
- PORT_DelModem - This function allows you to delete an existing modem configuration.
- PORT_DelPort - This function allows you to delete an existing port configuration.
- PORT_EditDisplay - This function allows you to edit the characteristics of an existing Display Configuration file.
- PORT_EditGlbSet - This function allows you to edit the characteristics of an existing Global Configuration file.
- PORT_EditModem - This function allows you to edit the characteristics of an existing modem configuration file.
- PORT_EditPort - This function allows you to edit the characteristics of an existing Port configuration file.
- PORT_Exit - This function allows you to exit the Port module
- PORT_ListDisplays - This function will list all available Display configurations.
- PORT_ListGlbSets - This function will list all available Global configurations.
- PORT_ListModems - This function will list all available modem configurations.
- PORT_ListPorts - This function will list all available Port configurations.
- PORT_ViewCompTypes - This function will list all computer types.

SIG

- SIG_Add - This function allows you to create a new SIG (special interest group).
- SIG_AddArea - This function allows you to add areas to an existing SIG.
- SIG_Del - This function allows you to delete an existing SIG.
- SIG_DelArea - This function allows you to remove areas from an existing SIG.
- SIG_Edit - This function allows you to edit the characteristics of an existing SIG.
- SIG_Exit - This function exits the SIG module.
- SIG_List - This function lists all available SIGs.
- SIG_ListAreas - This function lists all areas included in a given SIG.
- SIG_ToggleType - This function toggles the SIG editor between message and file area SIGs.

SYSUSER

- SYSU_Add - This function allows you to add a new user to the system.

SYSU_AddTplt	- This function allows you to create a new user validation and editing template. A template applies many attributes at once, some being upgrading, and some being downgrading attributes.
SYSU_Delete	- This function allows you to delete a user from the system. Depending on the size of your system, this could be a lengthy operation.
SYSU_DelTplt	- This function allows you to remove a template from your system. Templates are only used at the time you upgrade or downgrade a user's account, therefore, removing a template has no effect on your current users.
SYSU_Edit	- This function allows you to edit the account of a user, either by applying a template, or modifying individual items in their account.
SYSU_EditTplt	- This function allows you to edit an existing template.
SYSU_Exit	- This function exits the SysUser module.
SYSU_GlobalTplt	- This function allows you to apply a given template to all users within a given user-level range.
SYSU_List	- This function allows you to list all of the users on your system.
SYSU_Purge	- This function allow you to purge users who have not called into your system within a given amount of time.
SYSU_Validate	-This function will allow you to validate new users, by either editing their account, or applying a template, or both.

Local Variables

The following is a list of local %switch variables for Mess and File.

Mess:

%Msg_MsgNumber	- The number of the current message
%Msg_AreaNumber	- The number of the current message area
%Msg_AreaName	- The name of the current message area
%Msg_Direction	- The user's current reading direction ("Forward" or "Reverse")
%Msg_ThreadMode	- The user's current thread-reading status ("On" or "Off")
%Msg_TagRead	- The user's current tag-read status ("On" or "Off")
%Msg_LowMsg	- The number of the low message in the current message area
%Msg_HighMsg	- The number of the high message in the current message area
%Msg_SigNumber	- The number of the user's current SIG ("0" if no SIG selected)
%Msg_SigName	- The name of the user's current SIG ("NO SIG" if no SIG selected)
%Msg_UserHighMsg	- The user's high message pointer in the current message area
%Msg_AreaType	- The type of the current message area ("LOCAL", "NETMAIL",

"ECHOMAIL", "USENET", "PRIVATE")

- %Msg_Zone - The current port's FidoNet ZONE number
- %Msg_Net - The current port's FidoNet NET number
- %Msg_Node - The current port's FidoNet NODE number
- %Msg_Point - The current port's FidoNet POINT number
- %Msg_From - The name of the user the current message is FROM
- %Msg_UFrom - The underscored name of the user the current message is FROM
- %Msg_To - The name of the user the current message is TO
- %Msg_UTo - The underscored name of the user the current message is TO

File:

- %File_FileNumber - The number of the current file
- %File_AreaNumber - The number of the current file area
- %File_AreaName - The name of the current file area
- %File_LowFile - The number of the low file in the current file area
- %File_HighFile - The number of the high file in the current file area.
- %File_SigNumber - The number of the user's current SIG ("0" if no SIG selected)
- %File_SigName - The name of the user's current SIG ("NO SIG" if no SIG selected)
- %File_UserHighFile - The user's high file pointer in the current file area
- %File_FileName - The name of the current file
- %File_FilePath - The full path to the current file
- %File_From - The name of the person who uploaded the current file
- %File_UFrom - The underscored name of the person who uploaded the current file.

Index

Symbols

%Switches 105, 279
%switches, TrapDoor 252
0MinWarn.txt 109
2232 multi-serial 8
2MinWarn.txt 109
7WIRE 252

A

A2232 41, 42
ABORT 252
access level 134
ACCOUNTING 208, 253
ACCTMAX 253
ActivatePort 149
Activity string 134
ACTIVITYLOG 202
Add Area 46
Add User 54, 62
ADDRESS 198
AddTime 149
adduser 9
ADJUST 253
AFTERSESSION 254
AKA 199, 254
alias 55
ALLOWREFUSING 254
Alternate Path 97
AmigaDOS 1, 2
ANSI 32
ANSI Control Switches 280
ANSWER 208, 254
Answer Mode 28, 36
Answer String 27, 35
Application.txt 108
AppliedTemplate.batch 114
ARCADD 204
ARCEXTRACT 204
Archiver Settings 204
Archivers, approved 205
Area Capacity 59
Area Copy 62
Area Name 97
Area Number 97
area SysOps 55
AREAFIX 202
AreaInfo 150
ARexx, DLGMail 206
ARexx script 128
ARexx, TrapDoor 269

ARJADD 204
ARJEXTRACT 204
ARP 5
Auto Access Area 97
Auto-Access 16, 45, 46, 54, 57
AUTOCALL 199, 200
AUTOCALLZONE 201
AUTOOVERSCAN 254

B

backbone 212
BACKGROUND 255
Background 30, 40
Bad, message area 195
BADMSGSDIR 200
BANNER 255
Batch file 128
Batch Upload 100
BAUD 255
Baud Rate 25, 28, 36, 37
BaudRateTooLow.TXT 110
BBS 1
BBS Name 25, 36
BBS-Manual.txt 110
BBSCHAR 255
BBSCOMMAND 255
BBSINOUT 256
BBSMODE 256
BBSPORT 200
Bit-Planes 30, 39
BOSS 256
Broadcast 150
broadcast messages 134
BUFFERSIZE 240
Built-In Functions 281
bulletin board system 1
Bumping Filenames 276
bundle 213
bundles, packet routing 213
bundling, for points 233
bundling mail, examples 215

C

CALL 199, 207, 256
call accounting 246
CALLTYPES 199
Capacity 47
CatchUp 151
CD ROM
file areas, CD ROM tutorial 90
CD ROM, guidelines 91

- CD ROM, side effects 93
- Chain 133
- Change Areas 49
- Change SIG 49
- CHANGEFLOW 208, 223
- CHANGEPKT 208, 223
- Character Set 25, 37, 47, 59
- character translation sets 112
- Chat 29, 37, 113, 151
- CHECKCRC 240
- CLASS 211
- CLI mode 133
- CloseAreas 152
- Co-SysOp 11
- area SysOp 95
- colours 32, 257
- Command Mode 35
- Command Mode String 27
- Command stack 128
- command stacking 13
- commands
 - use in batch files 149
 - use in CLI 149
 - use in menus 149
- COMMENT 240
- CompileLang 153
- CompileScreen 119, 153
- Compressed Mail 273
- computer types 43
- concepts 11
- conferencing 13, 19
- CONFIG 257
- Configuration Files 118
- ConfScr 154
- ConfUser 154
- Connect Delay 25, 37
- CONT 210
- Cont Read 49, 67
- Copy Users 54
- Correct Msg 49
- CRASH 207, 222
- CRASHMAIL 199
- Crashmail.batch 116
- CREDITS 257
- CronEvent 154
- CronTab 118
- CRtoLF 157
- Custom Language 26, 38
- Custom log value 134
- Custom Origin Line 58
- Custom Screen 25, 36

D

- data compression 35

- DB_ALL 203
- DB_AREAFIX 203
- DB_BUNDLE 203
- DB_DLGMAIL 203
- DB_EXP 203
- DB_HUB 203
- DB_IMP 203
- DB_NET 203
- DB_TICK 203
- DeActivatePort 157
- Default Command Stack 26, 38
- Default Menu 25, 37
- DELAYCALL 199, 207, 222
- Delete All 49
- Delete User 54, 62
- DELOLDDIFFS 241
- DELOLDLISTS 241
- DF 157
- DFExport 157
- DIAL 241
- DIALER 258
- DIETIFNA 258
- DIRECT 207, 222
- DIRECTMAIL 199
- DIRECTONLY 199, 207, 222
- Display Configuration 28
- Display Control Switches 280
- Display File 38, 41
- DLG.Start 17, 33, 114
- DLGBundle 212, 231
- DLGBundle, executable 194
- DLGEdit 158
- DLGExp 228
- DLGExp, executable 194
- DLGImp 210, 227
- DLGImp, executable 193
- DLGMail 190, 193, 209
 - communication 221
 - configuration 233, 235
 - executable 193
 - installation 195
 - interfacing with DLG 193
 - interrupting 232
 - macros 224
 - mail processing commands 222
 - nodelist commands 223
 - overview 221
 - path assignments 195
 - required archivers 226
 - Scripts 225
 - structure 193
 - TrapDoor commands 223
 - using 206
- DLGMail.ARE 118, 209

- DLGMail.ARE, in use 211
- DLGMail.BUN 118, 214
- DLGMail.CFG 118, 197
- DLGMail.MAC 118, 224
- DLGMail.MRT 118, 216
- DLGMail.PNT 118, 218
- DLGNet 212, 230
- DLGNet, configuration 211
- DLGNet, executable 194
- DLGNet, routing 212
- DLGUUSend 158, 191
- DLGXmodem 158
- DLGZmodem 159
- DMC change commands 220
- DMC, executable 193
- Door 159
- download 76
- downloading 15
- DrivelsFull.Batch 115
- Drop To DOS 20, 120
- DSR 258
- DTR 28, 36

E

- Echo.txt 109
- EchoMail 47, 58, 189
- echomail, for points 233
- EchoMail, hub request 238
- EchoMail, message area 195
- Edit File 100
- Edit Signature 49
- Edit User 54, 62
- EMSI 258
- Enter Msg 48, 64
- EnterArea 160
- EnterArea.txt 98, 110
- equipment 5
- Event.txt 109
- Executable program 127
- EXIT 256
- EXPORT 208, 222, 223

F

- features 1
- Fido:, path assignment 195
- FidoConfig 160
- FidoNet 11, 33, 56, 189, 245
 - joining 189, 195
 - requesting address 237
 - setting up first time 235
 - setup 194
- FidoStart.batch 116
- FILE 101
- File 160

- File Area Maintenance Editor 99
- file areas 12, 96
 - access privileges 97
 - associating with message areas 94
 - auto-access 94
 - control files 96
 - creation 71
 - deleting tutorial 93
 - maintenance 86
 - pay system 95
 - philosophy 72
 - special access 95
 - structure 94
 - text and batch files 98
 - types 71
- file base 18
- File Request 97
- file transfer protocol 15
- File.DAT 96
- FileAreas 161
- FileMaint 161
- FilePurge 162
- Files
 - batch uploading 88
 - checking for illegal uploads 95
 - commenting 82
 - comments 79
 - descriptions 81
 - downloading tutorial 85
 - editing 83
 - editing description 80
 - enforced upload/download ratios 95
 - listing 77
 - isting archive 80
 - moving 98, 99
 - partial uploads 103
 - tagging 79
 - uploading tutorial 81
 - validation 81
- FileSearch 162
- FileUsers 162
- Filter 49, 67
- FinishedApplication.txt 108
- FixUsers 162
- FKEY 259
- FLOATLOCK 259
- Flow files 214, 273
- Font 29, 30, 40
- Forced Command Stack 26, 38
- Forward 163
- Forward Msg 49, 66
- Forward Read 49, 67
- FreeMisc 163
- FreePort 163

FREQUEST 259
Freshen 100, 163
full screen editor 51, 64

G

GCFG 209
GenConfig 163
General Information Switches 280
GENNEWSTYLE 241
GetMisc 164
GetPort 164
GETTY 113
Global Configuration 24, 36
Global Edit 100
global new scan 15
global newscan areas 47
global paths 92
Global Screen 25
Global Set 41
GoodBye 165
Group 165
group accounts 69
grouping related message areas 55

H

Handles 47, 55, 58
handles, EchoMail
aliases, EchoMail 58
Hang-up String 27, 35
HangUp 165
HappyBirthday.txt 109
hard drive partition 7
Hayes 34
Header Scan 50, 68
HELP 206
Help file 133, 134
Hi-Res 30, 40
High Message Pointer 50, 68
hot keys 14
HUB 203

I

IB 116
Idle Timeout 25, 37
Immed 166
IMMEDIATE 259
INBOUND 260, 274
Init String 27, 34
installation 6
INTERLACE 260
Interlace 30, 40

J

Jive 60, 167

K

keyboard commands, TrapDoor 248
Kill Msg 48, 65
KilledFiles 96
KillTD 116
Kraut 60, 167

L

language 13
language files 111
%switches 112
editing 111
LastBoot.txt 111
LeaveArea 167
Lex 167
Lex Check 49, 66
LFtoCR 167
LHAADD 204
LHAEXTRACT 204
line editor 64
LineEdit 168
LINK 211
LINKREPLIES 202
List Area 54, 62
List Readers 50, 69
List Users 54, 62
ListPorts 168
ListUser 168
local 17, 23, 34, 114
LocExtEditor 114
LOCK 260
Lock Mode 27, 35
LOG_ALL 203
LOG_AREAFIX 203
LOG_BUNDLE 203
LOG_DLGMAIL 203
LOG_EXP 203
LOG_HUB 203
LOG_IMP 203
LOG_NET 203
LOG_TICK 203
LOGFILE 260
Login.batch 114
Login.DLGBatch 114
Login.txt 108
LOGLEVEL 261
LOGLINES 202
Logout.DLGBatch 114
Logout.txt 109
Logs:, path assignment 196
LOGWINDOW 261
LZHADD 204
LZHEXTRACT 204

M

- mail processing 219
- Mail:, path assignment 196
- Mailer 245
- MailPack 168
- MAIN MENU 12, 18
- maintaining message area continuity 55
- Maintenance.batch 116
- MakeAlias.Batch 116
- MakeFList 169
- MAXBAUD 262
- Menu 11, 48, 170
- menu item, hidden 134
- menu item, priority 134
- Menu Script 128
- menus 13, 121
 - %switches 124
 - built in 121
 - built-in functions for File 131
 - built-in functions for Mess 129
 - chain 133
 - configurable 121, 125
 - configurable adding item 127
 - configurable columns 127
 - configurable menu entry type 127
 - configurable module name 127
 - configurable prompts 127
 - configurable title 126
 - custom example 122
 - custom file format 122
 - custom menu tutorial 123
 - custom sets 121
 - file naming conventions 121
 - help file 133
 - local variables for File 135
 - local variables for Mess 134
 - menu sets and text files 125
 - numeric entry 127
 - overlay 133
- Mess 63, 170
 - command line arguments 63
 - directing a SIG choice 63
 - enforcing a SIG choice 63
 - forcing an area choice 63
 - inserting a command stack 63
- Message Area Commands 64
- message area pointers 69
- Message Area Reference 56
- message area statistics 48
- message area user access 62
- Message Area User Editor 53
- Message Areas 12, 45
 - FidoNet 194
 - user activity key levels 57

- message directories and files 56
- message exporting 211
- Message Filters 61
- message pointers, editing 56
- Message, The 212
- message tossing 211
- MINBAUD 261
- modem commands 32
- Modem Configuration 26, 34
 - TrapDoor 249
 - TrapDoor examples 250
- Modem File 41
- MODEMANSWER 262
- MODEMDIALPOST 262
- MODEMDIALPRE 262
- MODEMHANGUP 263
- MODEMINIT 263
- modular 1
- MSG:, path assignment 196
- MSG_AREA_# 209
- MSGADDED 206
- MSGAreas 170
- MSGUsers 171
- multi-line 33
- MYADDRESS 209
- MyQuotes.quote 110

N

- NAME 263
- navigate 12
- NetMail 47, 58, 189
- netmail, for points 233
- NetMail, message area 195
- netmail routing, examples 217
- NetMail, sending replies 70
- Netmail, sending to private node 70
- NETMAILDIR 200
- NETMAILROUTE 202
- NETSCAN 208, 223
- networked mail 45
- NewsGroup 190
- NewUser 171
- NewUser.batch 114
- NewUser.DLGBatch 114
- NewUser.txt 108
- Next Area 49
- NLComp.DMB 116
- NLDiff.DMB 116
- NLNew.DMB 116
- NOANSWER 208
- NODE 263
- Nodediff 239
- NodeInfo 171
- NODELIST 207, 241, 264

- Nodelist 239, 245
- nodelist 190
- NodeList compiling 239
- NODELIST DIFF 223
- NODELIST NEW 223
- NODELIST RECOMPILE 223
- NODELISTPATH 241
- NOLINK 211
- NONE 256
- NOPASSTHRU 210
- NoProceed.txt 109
- NoRegistration.txt 109
- NORMAL 207
- NORMALMAIL 199
- NOSTRIP 210
- NotEnoughTime.txt 109
- NOTIFYACTIVEONLY 201
- NTSC 29, 38, 39, 41

O

- OpenAreas 171
- OPERATOR 199
- ORIGIN 199
- OUTBOUND 264
- Overlay 133

P

- Packet files 273
- packet, name extensions 213
- packet routing 213
- packet routing, examples 216
- packet, The 212
- packets, 4D and 2D naming 213
- PAL 29, 38, 39, 40, 41
- Pass:, path assignment 196
- PASSTHRU 210
- PASSWORD 242, 264
- password 9
- passwords, TrapDoor 247
- Pausing 25, 37
- pay system 55
- PDQAreaFix 211
- PDQAreafix 219
- PDQMail.AFX 118
- PKTBUFFERLENGTH 201
- PKTRoute 202
- point routing 218
- Point Size 40
- POINTENABLE 201
- Pointers.FILE 96
- Pointers.MSG 56
- POINTNET 199
- points, routing 234
- points, servicing 233

- PolyXArc 205
- Pop Global Screen 36
- Port 11, 171
- port 34
- Port Configuration 23, 34, 41, 43
- port configuration 30, 34
- PortInfo 172
- PortIsPrivate.TXT 110
- Ports 30
- Post Bulletin 48, 65
- Post-Dial.DMB 117
- Post-DLGBundle.DMB 117
- Post-DLGNet.DMB 117
- Post-Export.DMB 117
- Post-Import.DMB 117
- Post-Tick.DMB 117
- Pre-Dial.DMB 117
- Pre-DLGBundle.DMB 117
- Pre-DLGNet.DMB 117
- Pre-Export.DMB 117
- Pre-Import.DMB 117
- Pre-Tick.DMB 117
- PrepAreas 172
- Private Mail 49, 66
- private mail, monitoring 55
- private message area 45
- PROCESS 208, 222
- process cycle 218
- PROCESS_HEIGHT 198
- PROCESS_WIDTH 198
- PROCESS_X 198
- PROCESS_Y 198
- PROCESSWINDOW 198
- Public 25, 37
- PUBSCN_LACE 197
- PUBSCN_NAME 197
- PUBSCN_OSCAN 197
- PUBSHELL 206

Q

- quick tour 17
- QUICKATTACH 207, 224
- QUICKLOG 201
- QUICKREQUEST 207, 224
- QUIET 264
- QUIT 206, 224, 264
- Quote 172

R

- RAW mode 133
- Read Original 50, 68
- Read Reply 50, 68
- ReadLog 173
- ReceivedFile.batch 98, 114

- RECONFIG 206, 207, 223
- REDIALDELAY 265
- RefuseDownload.txt 98, 109
- remote 34
- RemovePort 173
- RemoveUser 173
- Renumber 47, 59, 174
- Reply To Msg 48, 64
- REPORT 206, 224, 242
- REQUEST 207, 222
- Request files 274
- ReRoute Private Mail 26, 38
- Reset String 27, 35
- ResourceReport 174
- RETRIES 265
- Reverse Read 49, 67
- REXXNAME 265
- Ring String 27, 35
- RINGS 265
- routing, examples 216
- routing, wildcards 214
- RTS/CTS 35
- RUN 265

S

- SAVEDAMAGED 202
- Screen 29, 39
- Screen.DAT 110
- Screen.MSG 110
- Screen.msg 118
- ScreenEdit 174
- ScreenEdit.help 110
- SCREENMODE 265
- script files 113
- Seen-By 58
- SendMsg 175
- Serial Device 41
- serial.device 41
- SERIALFLAGS 266
- SERIALNAME 266
- SERIALUNIT 266
- SetMenuPri 176
- Setup 177
- SHARED 266
- Shell 17
- SHOW_AREAFIX 204
- SHOW_BUNDLE 204
- SHOW_DLGMAIL 204
- SHOW_EXP 204
- SHOW_HUB 204
- SHOW_IMP 204
- SHOW_NET 204
- SHOW_TICK 204
- SHOWREXX 266

- SIG 65
- Signature 65
- Signatures 47, 58
- signatures in NetMail 69
- Sigs 177
- Skip Thread 50, 68
- SLOWMODEM 267
- smart input 14
- Spare(n) 115
- SPAWN 256, 267
- special access 16, 45, 54, 56, 57
- STACKSIZE 200
- startup-sequence 17
- STATUS 206, 267
- STATUS_WIDTH 198
- STATUS_X 197
- STATUS_Y 197
- STATWINDOW 268
- STRIP 210
- Sub menu 128
- SUBDIRECTORY_NAME 209
- super-high resolution 32
- Supra 30, 34, 41, 42
- SWEPULSE 268
- SysMenu 177
- SYSOP 268
- SysOp 11
- SysOp maintenance 13
- SysOp Menu 23
- SystemInfo.batch 115
- SysUser 177

T

- Tag Read 50, 68
- TAGNAME 209, 210
- TASKPRI 200, 268
- TBAud 178
- TColors 178
- TCont 178
- TDevQuery 178
- TDIH 178
- TDPORT 200
- TDSPAWN 201
- TempUploads 96
- terms 11
- TESTFREQ 268
- text and batch files 105
- text files 107
- batch files 98
- TFlags 179
- TFreeze 180
- Thread On/Off 50, 67
- thread reading, side effects 67
- TICK 202

- Tick.DMB 117
- Title.[port] 108
- Title.txt 108
- TKill 180
- TLx.startup 113
- Today.<month> 111
- TPT_Handler 14
- TPTBC 180
- Tptconf 180
- TPTCron 14, 118, 180
- TPTFreq 182
- TPTFreq.lst 119
- TPTQuote 182
- TPTRM 182
- TPTScript 113
- TPTShell 182
- TPTShell.cfg 120
- Transfer.batch 115
- TransferPort 183
- Translation 60
- Translator 47, 59
- TRAPDOOR 207
- TrapDoor 190
 - call accounting 253
 - custom configuration 246
 - example configuration 270
 - FidoNet compatibility 246
 - installation 245
 - interface with DLGMail 194
 - introduction 245
 - keyboard commands 248
 - license agreement 277
- TRAPDOOR ANSWER 223
- TRAPDOOR NOANSWER 223
- TRAPDOOR ON 223
- TrapDoor, refusing 248
- TrapDoor, security 247
- Trapdoor Setup 249
- TrapDoor.cfg 120
- TRAPDUMP 206
- TRAPDUMP OFF 223
- TRAPDUMP ON 223
- TrapList 194, 239
- Traplist configuration 240
- TrapList, example configuration 242
- TrapList.cfg 120
- Trecover 183
- TRIMLOGS 208
- TRx.startup 113
- TScreen 183
- TStat 184
- TString 184
- TTimeDelay 184
- TTitle 184

- Turbo Upload 99
- TWindow 184
- TWinHeight 185

U

- U.S. Robotics 34
- Undo 205
- UNet2DLG 185, 191
- Unit Number 41
- UpdateAreas 185
- upgrade 8
- upload 75
- Upload1.batch 99, 115
- Upload2.batch 99, 115
- UploadFile.txt 110
- Uploadfile.txt 98
- uploading 15
- uploads, group accounts 103
- USEADDRESS 210
- UseNet 11, 33, 47, 56, 58, 190
- User 11
- User Access to Message Areas 61
- user account 12, 15
- User Level 46
- User Personal Information Switches 279
- user-level 15, 54, 57
- user-options 19
- User-Startup 17
- User.FILE 96
- User.MSG 56
- USEREXPORT 208, 222
- UserInfo.txt 111
- UserOptions 186
- Utility Menu 20
- UUCP 33, 190
- UUCP Client 191
- UUCP client 113
- UUCP.batch 113
- UUCP2DLG 186, 191

V

- ValidatedUser.batch 115
- Validation 97
- Validation Area 97
- ValSpeak 186
- Valspeak 60
- VERBOSITY 201
- VERY IMPORTANT 200
- ViewArchive 187
- ViewArchive.batch 115
- visually impaired 33

W

- WaitPort 187

WAZOO 269
WHEREIS 206
Who 187
Window 29, 38
WMail 187
Workbench 1.3 5
WRAPLINES 269
WriteLog 187

X

XPRtransfer 187

Y

no entries

Z

ZEDZAP 269
ZIPADD 204
ZIPEXTRACT 204
ZMH 206, 256
ZMH OFF 223
ZMH ON 223
ZMHOff.DMB 117
ZMHOn.DMB 118
ZONE 242
zone gating 217
ZOOADD 204
ZOOEXTRACT 204

THE JOURNAL
OF THE
ROYAL ANTHROPOLOGICAL INSTITUTE
OF GREAT BRITAIN AND IRELAND
VOLUME 13
PART 1
1903

THE JOURNAL

OF THE

ROYAL ANTHROPOLOGICAL INSTITUTE
OF GREAT BRITAIN AND IRELAND
VOLUME 13
PART 1
1903